Who are still training GANs in 2025?

Jun-Yan Zhu
Carnegie Mellon University

Who are still training GANs

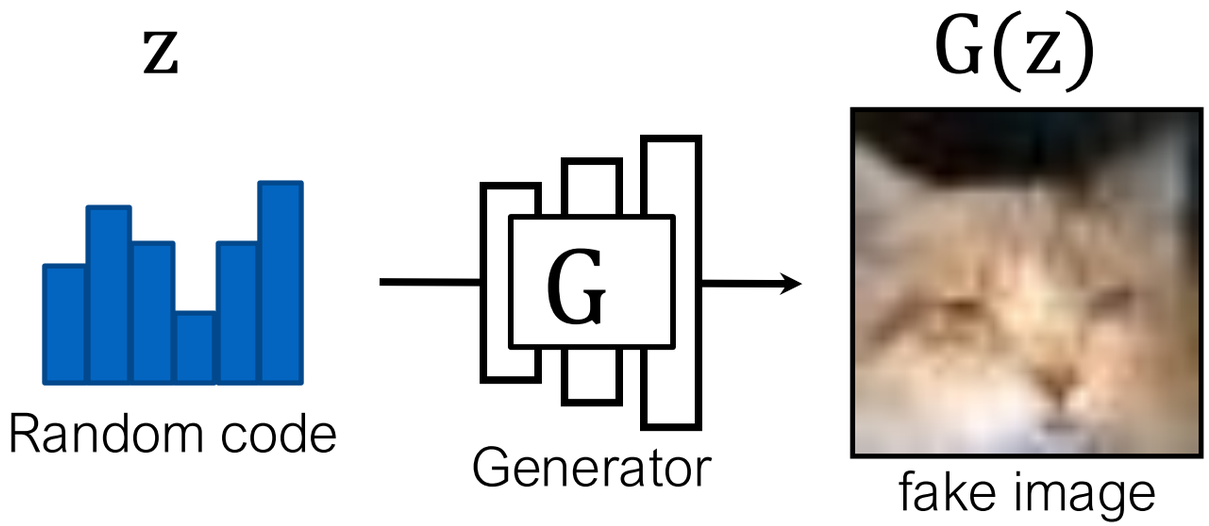
Short answer: our lab

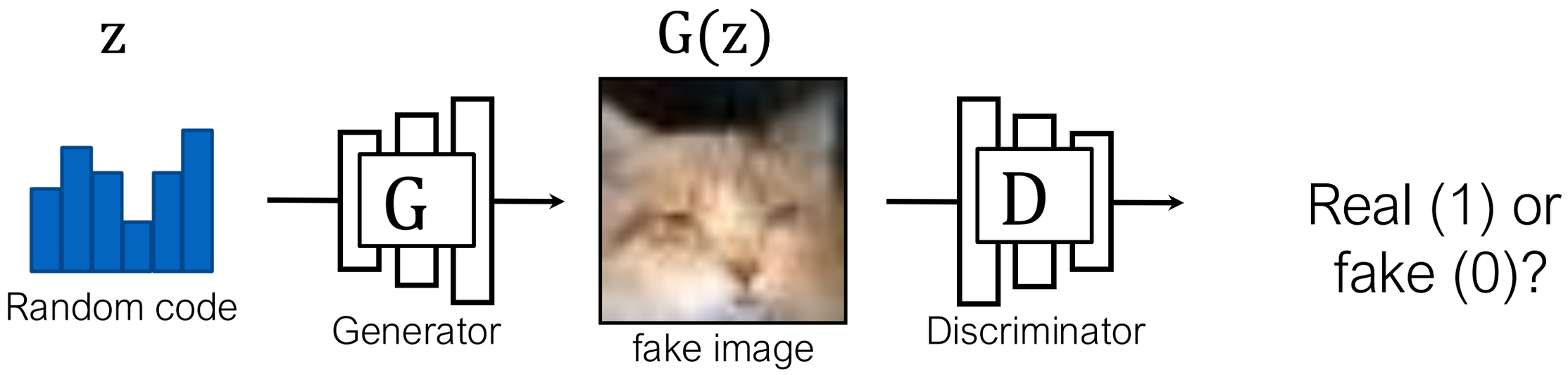
A more serious answer:
NVIDIA, Adobe, Meta, Stability AI (Stable Diffusion), BlackForest (Flux), ByteDance, …

Notes:

- People are training GANs without talking about them.
- Sometimes mention it in the implementation details and supplementary material.

GAN basics

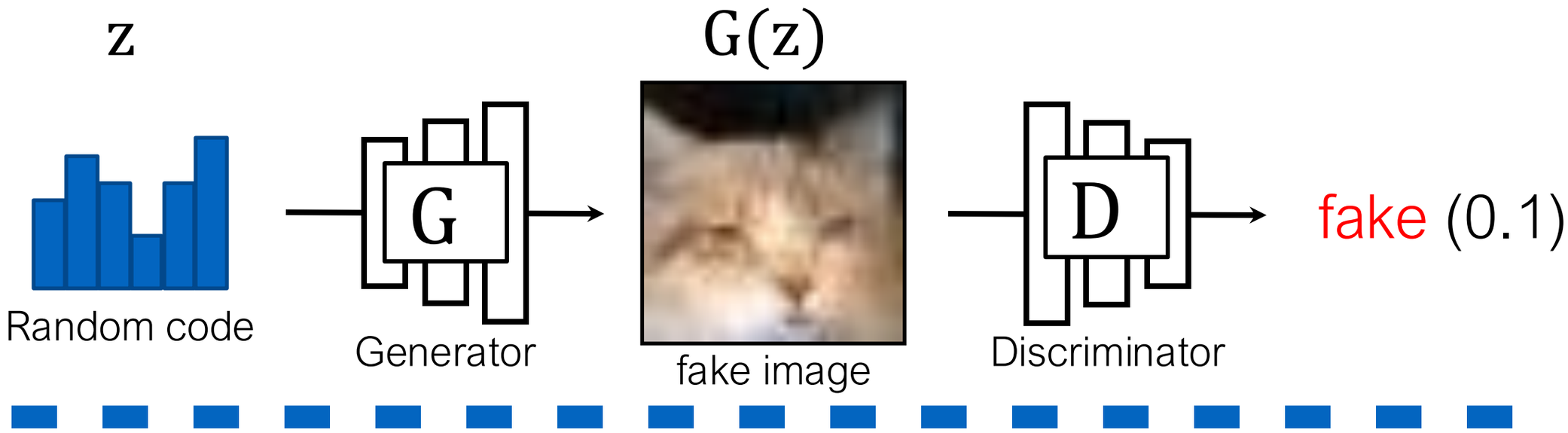

z
Random code
G
Generator
G(z)
fake image



A two-player game:

- G tries to generate fake images that can fool D.
- D tries to detect fake images.

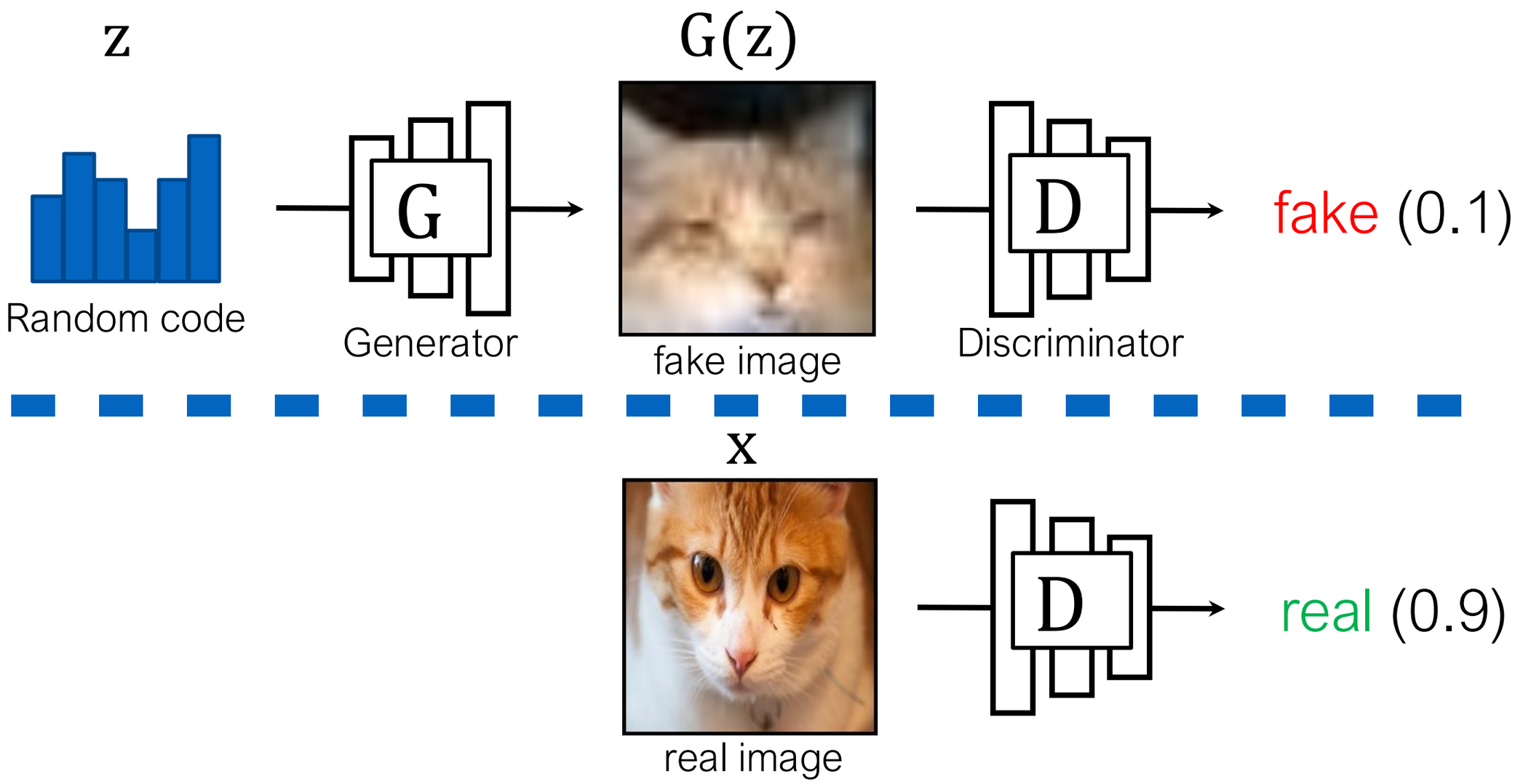
[Goodfellow et al. 2014]



Learning objective (GANs)

$$\min_G \max_D \mathbb{E}_z[\log(1-D(G(z))]$$

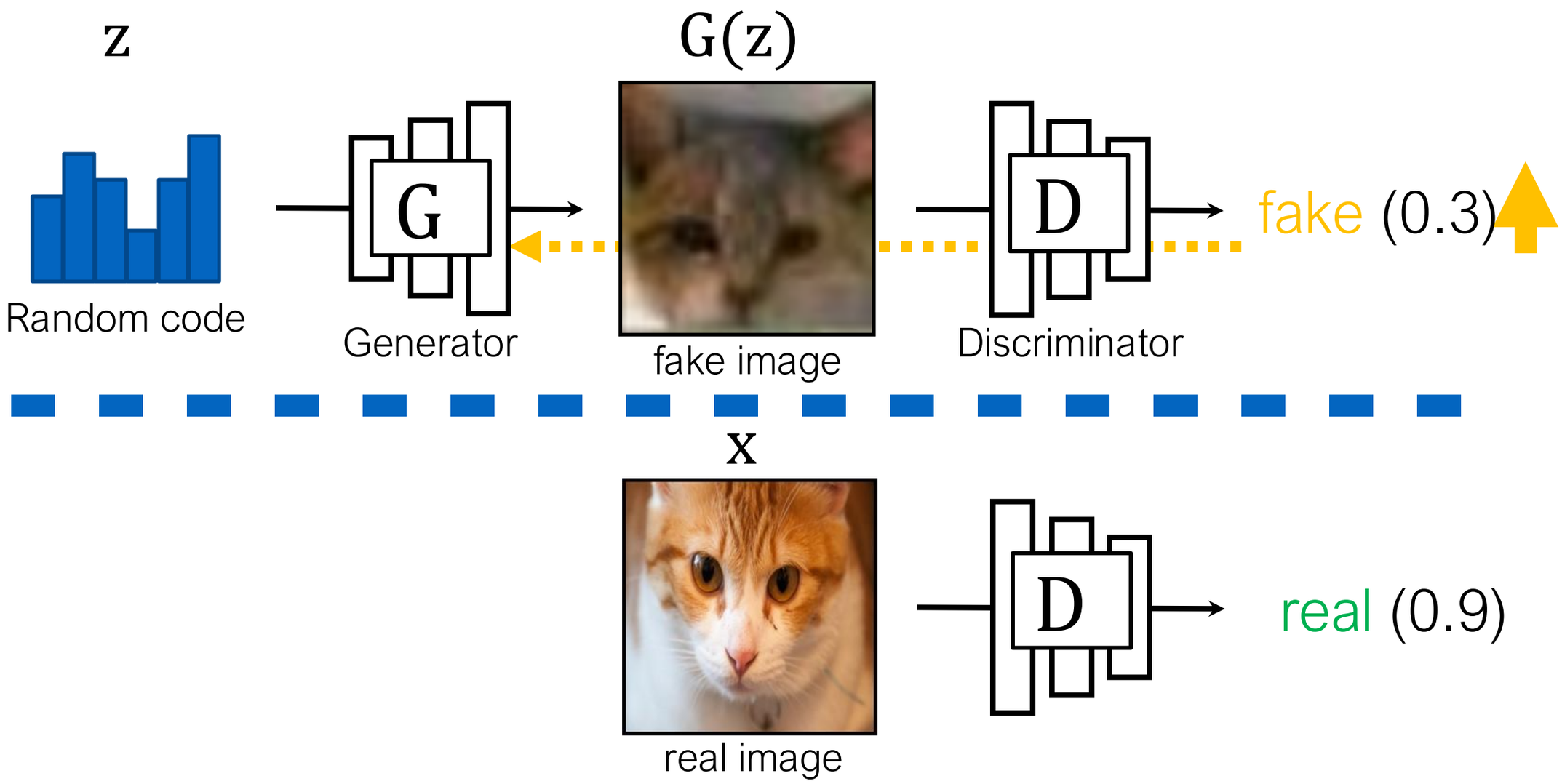
[Goodfellow et al. 2014]



Learning objective (GANs)

$$\min_G \max_D \mathbb{E}_z[\log(1-D(G(z))] + \mathbb{E}_x[\log D(x)]$$

[Goodfellow et al. 2014]

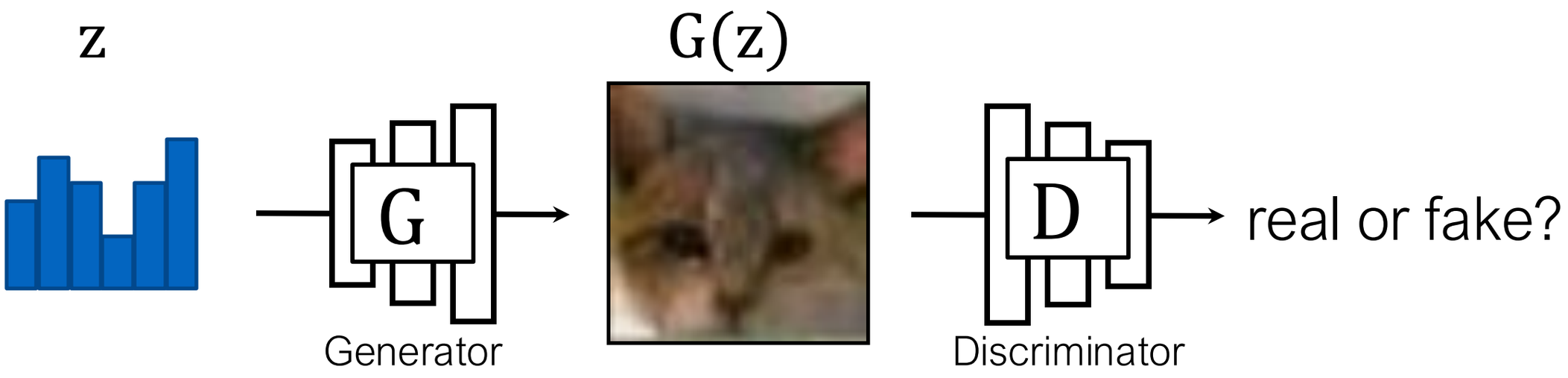


Learning objective (GANs)

$$\min_G \max_D \mathbb{E}_z[\log(1-D(G(z))] + \mathbb{E}_x[\log D(x)]$$

[Goodfellow et al. 2014]

GANs Training Breakdown



G tries to synthesize fake images that fool **D**

D tries to identify the fakes

- Training: iterate between training D and G with backprop.
- Global optimum when G reproduces data distribution.

[Goodfellow et al., 2014]

What has driven GAN progress?

Ian Goodfellow @goodfellow_ian · Jan 14

4.5 years of **GAN progress** on face generation. arxiv.org/abs/1406.2661 arxiv.org/abs/1511.06434 arxiv.org/abs/1606.07536 arxiv.org/abs/1710.10196 arxiv.org/abs/1812.04948

What has driven GAN progress?

Samples from **StyleGAN2 [Karras et al., CVPR 2020]**

Data

Data alignment

- Work well for well-aligned objects and landscapes.

(a) (b) (c) (d) (e) (f)

Figure 8: Creating the CELEBA-HQ dataset. We start with a JPEG image (a) from the CelebA in-the-wild dataset. We improve the visual quality (b,top) through JPEG artifact removal (b,middle) and 4x super-resolution (b,bottom). We then extend the image through mirror padding (c) and Gaussian filtering (d) to produce a visually pleasing depth-of-field effect. Finally, we use the facial landmark locations to select an appropriate crop region (e) and perform high-quality resampling to obtain the final image at 1024×1024 resolution (f).

See more details in Appendix of Progressive GANs [Karras et al., ICLR 2018]

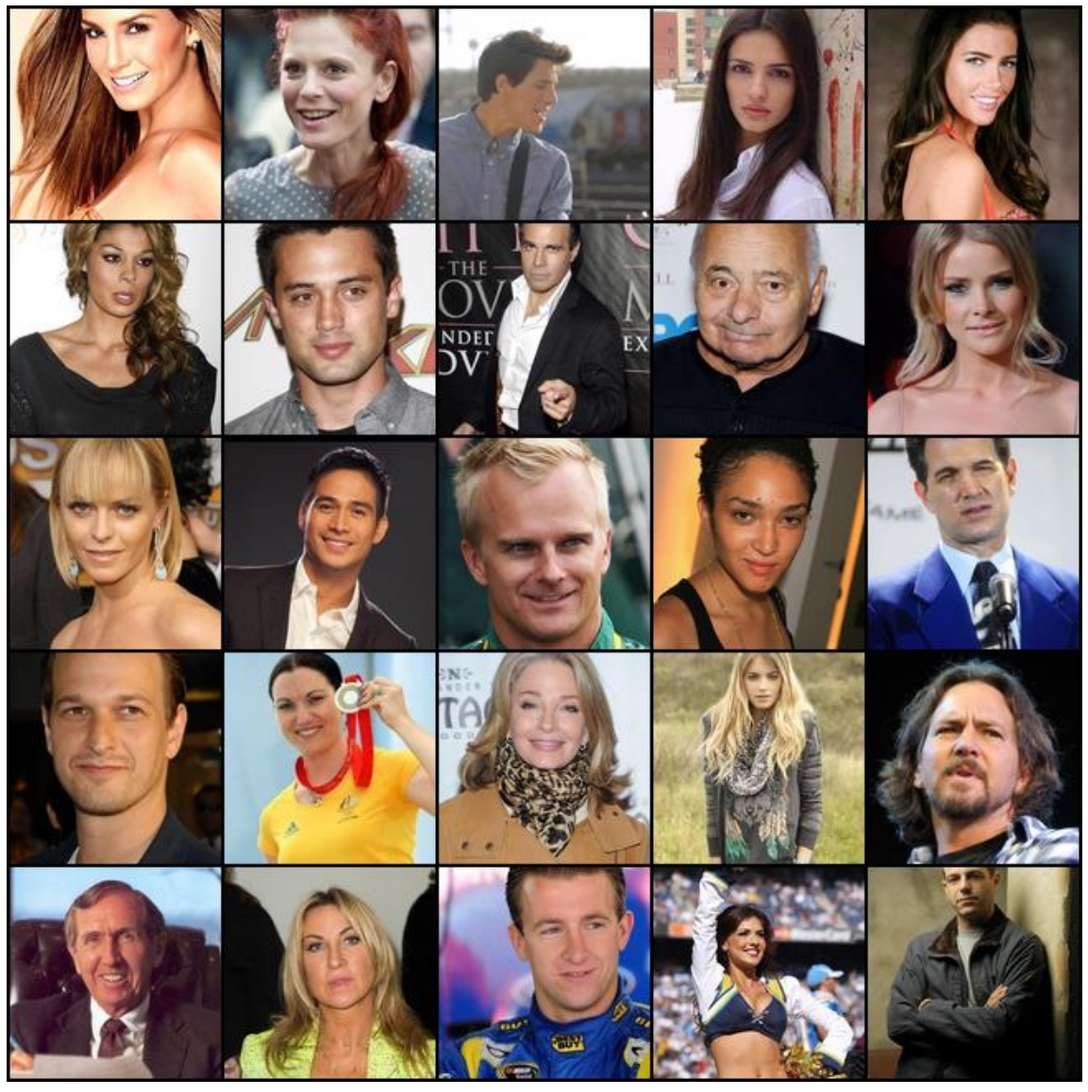
Aligned vs. unaligned data

Real images from aligned FFHQ

StyleGAN2 samples

Photo credit: StyleGAN2 [Karras et al., CVPR 2020]

Aligned vs. unaligned data



Real images from unaligned CelebA

StyleGAN2 samples

Photo credit: William Peebles (UC Berkeley)

Data are Expensive

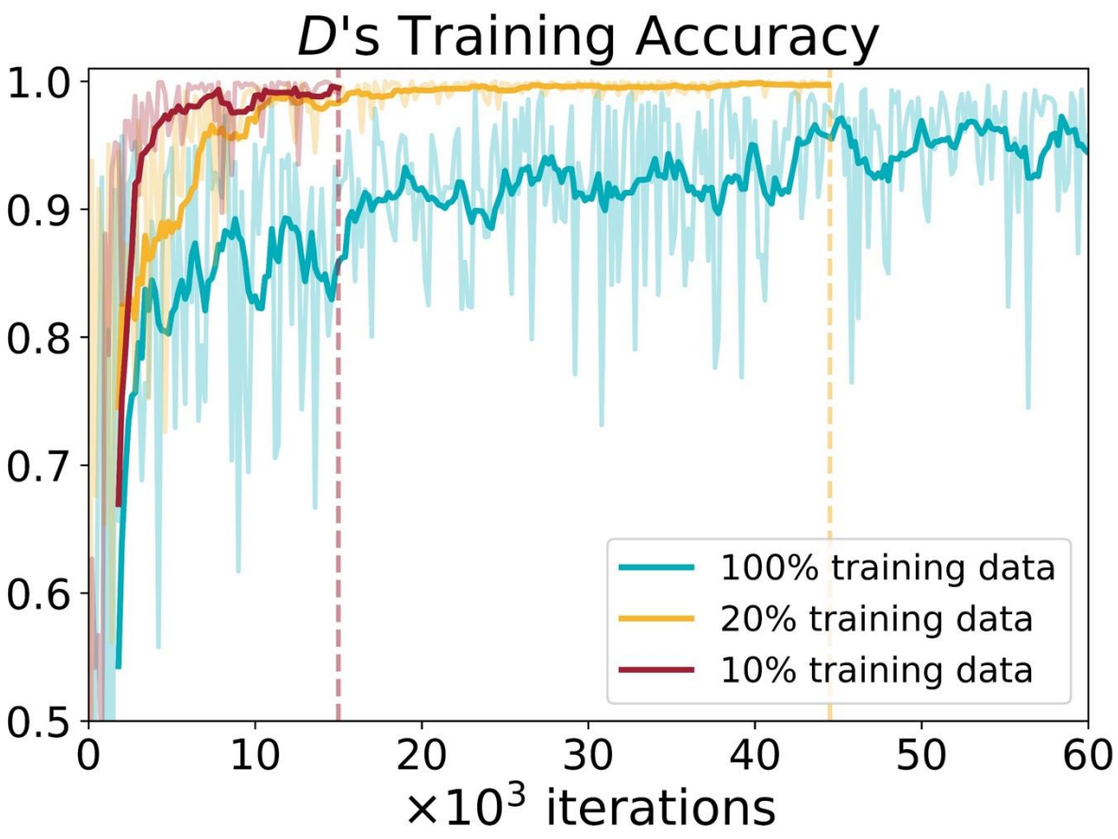
FFHQ dataset: 70,000 selective post-processed human faces ImageNet dataset: millions of images from diverse categories

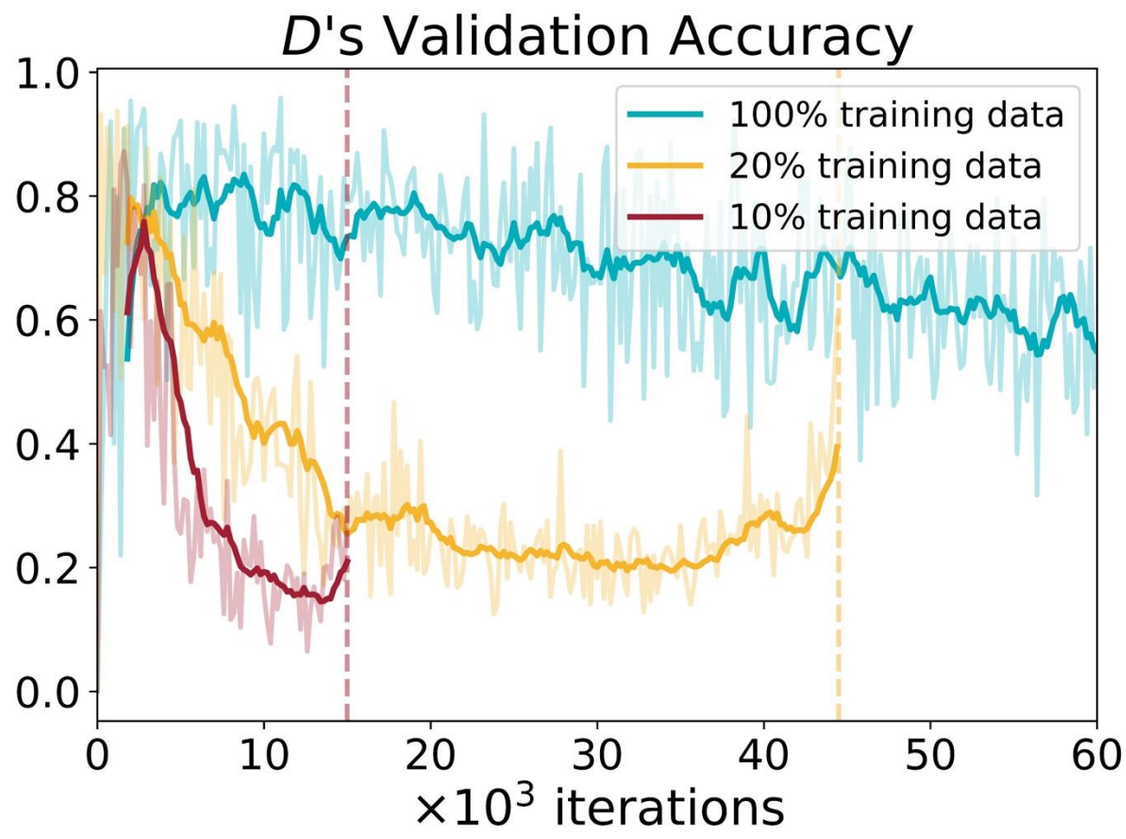
Months or even years to collect the data,
along with *prohibitive* annotation costs.

GANs Heavily Deteriorate Given Limited Data

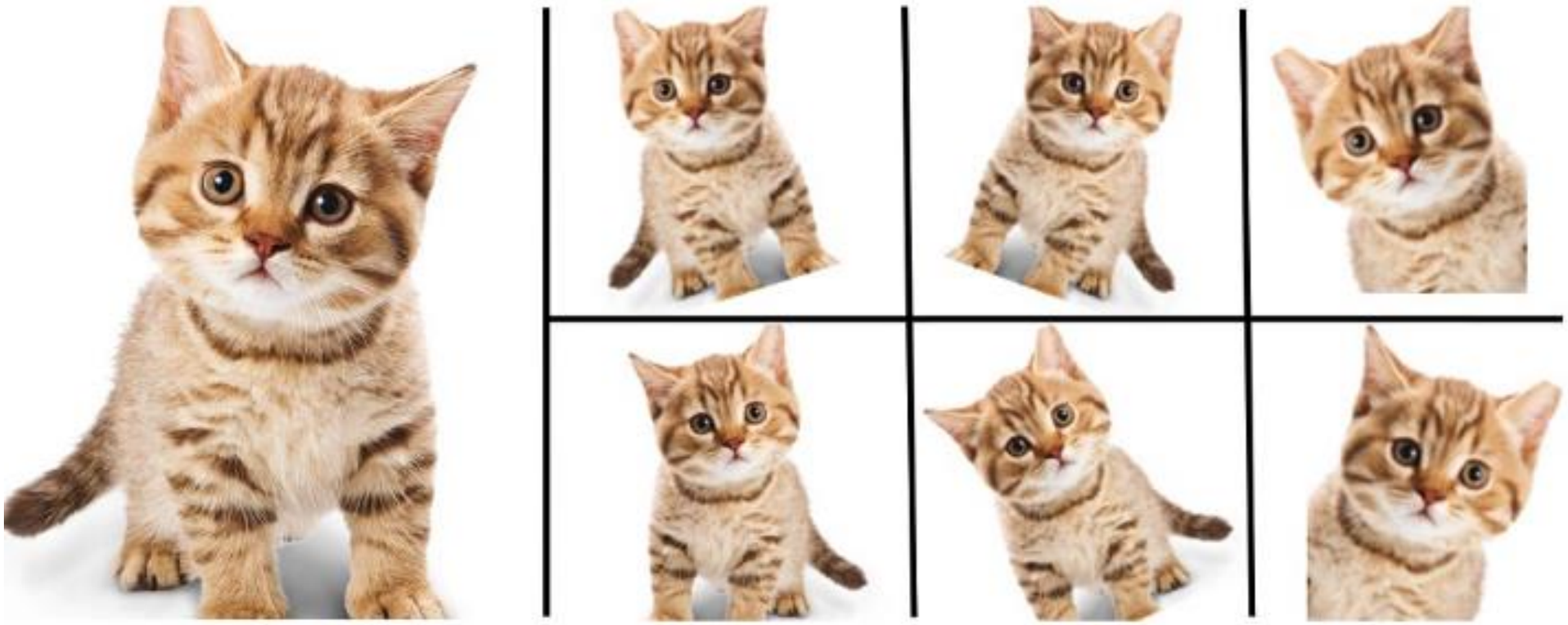
Generated samples of StyleGAN2 (Karras et al.)
using only hundreds of images

Discriminator Overfitting





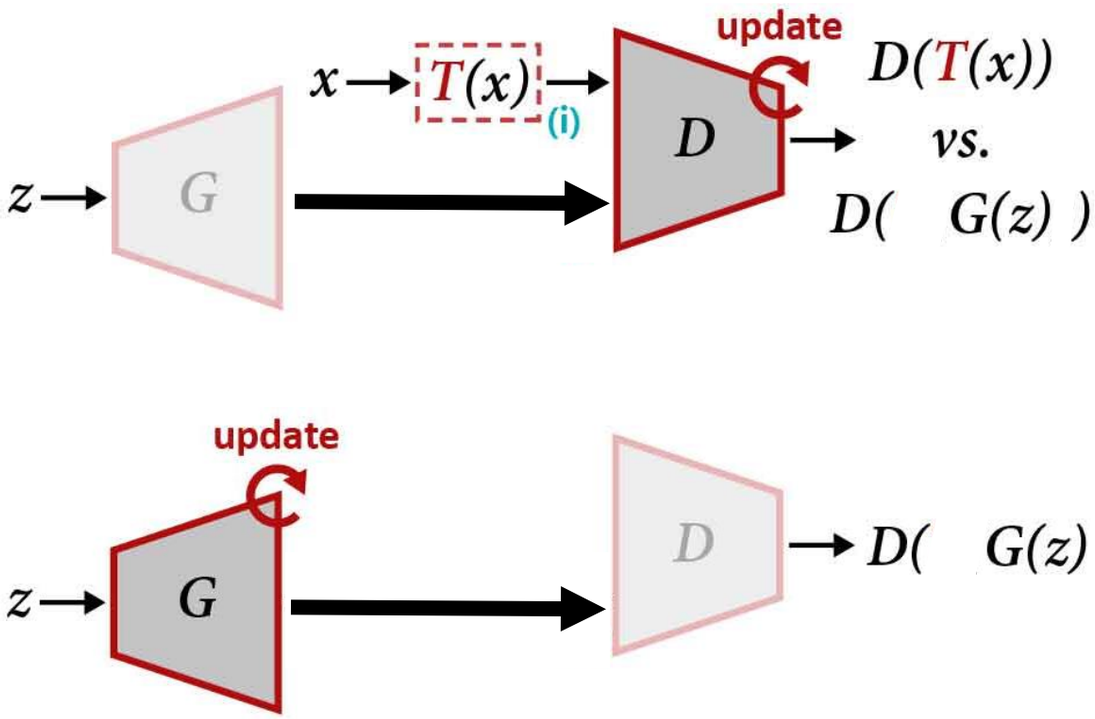
Data Augmentation



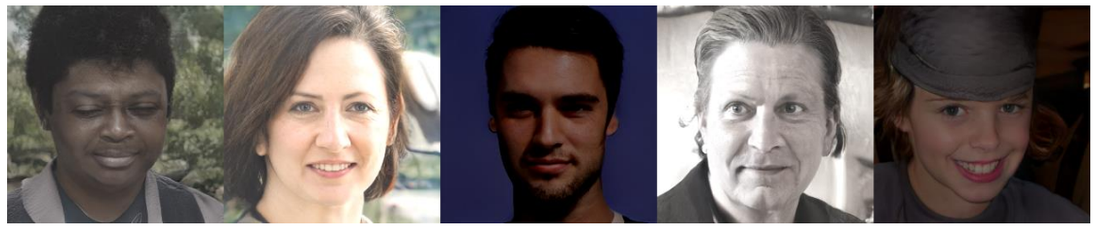
Data augmentation: enlarge datasets without collecting new samples.

How to Augment GANs?

#1 Approach: Augment reals only



Generated images



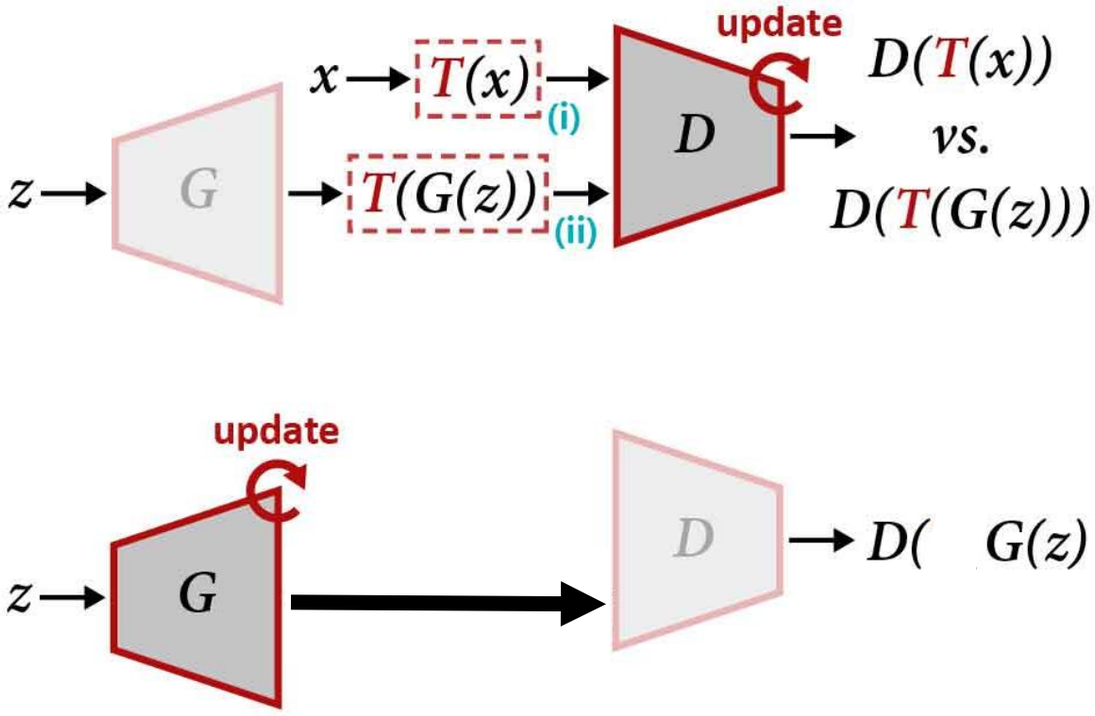
Artifacts from Color jittering

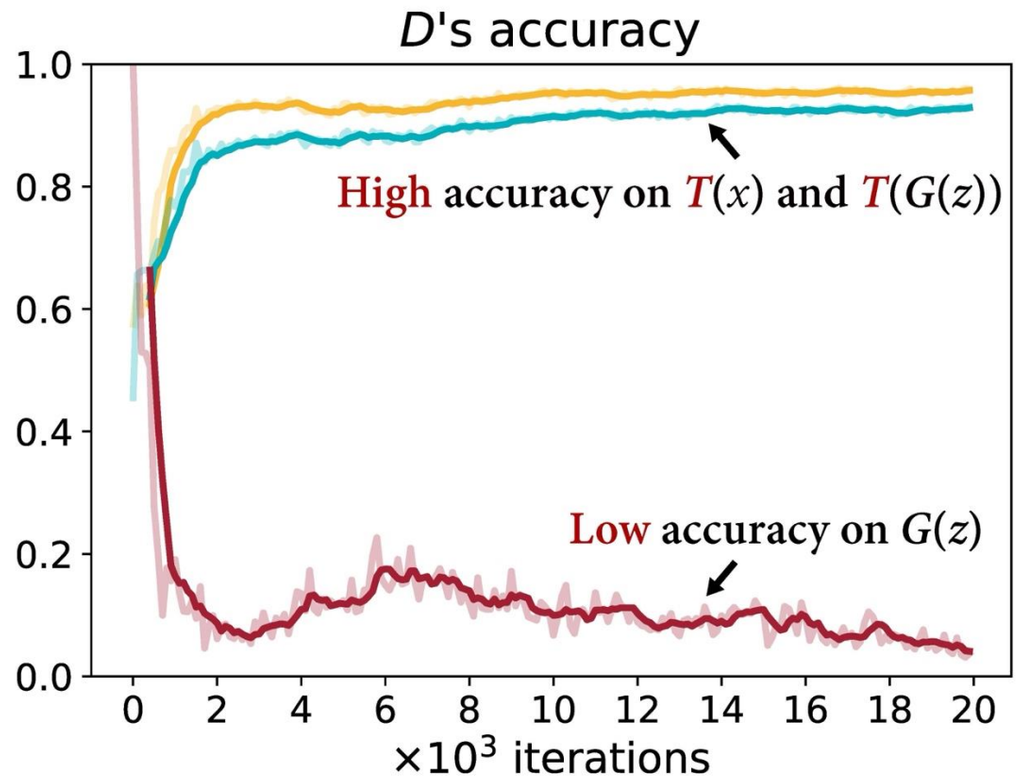
Artifacts from **Translation**

Artifacts from Cutout (DeVries et al.)

Augment reals only: the same artifacts appear on the generated images.

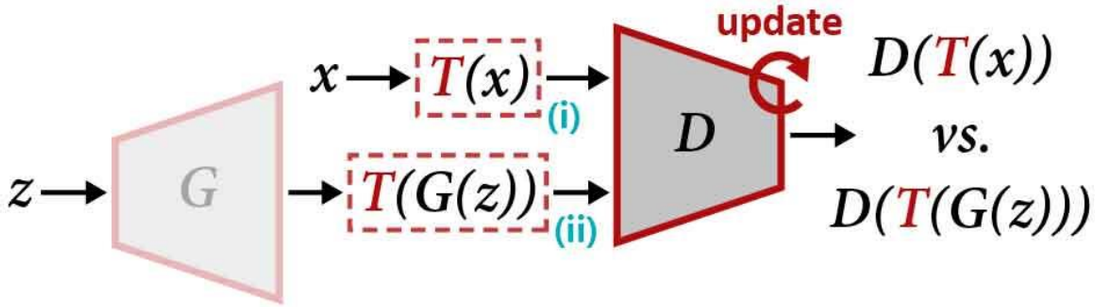
#2 Approach: Augment reals & fakes for D only

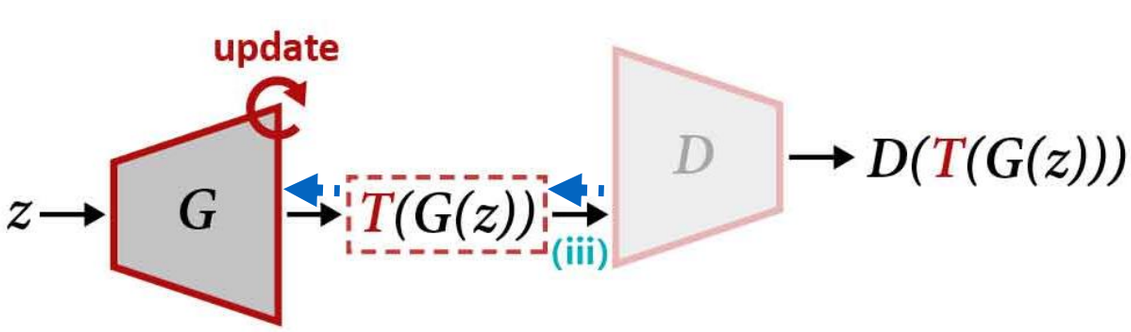


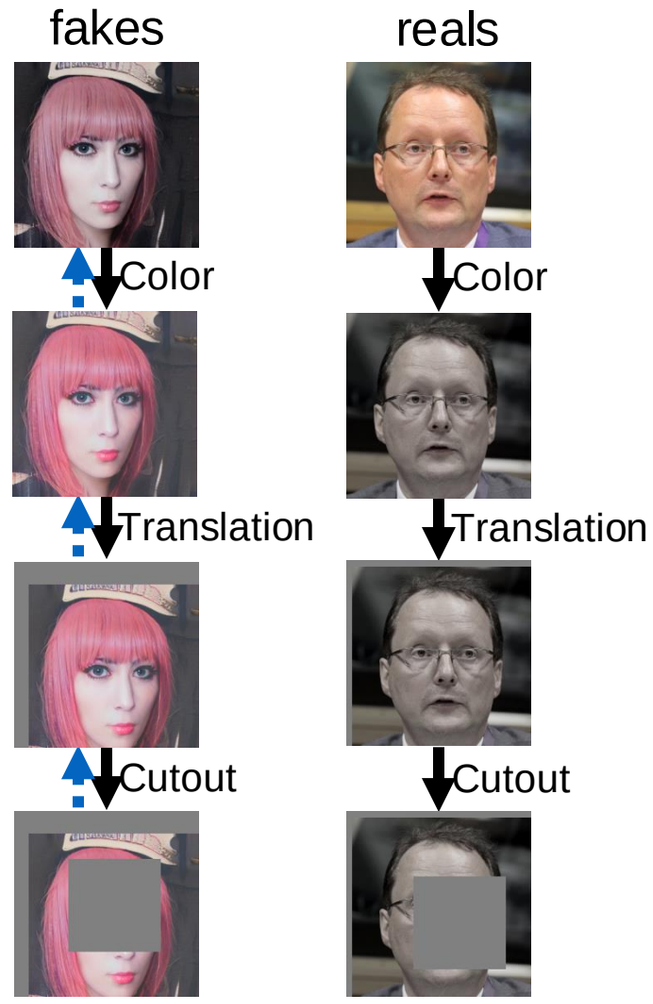


Augment D only: the unbalanced optimization cripples training.

#3 Approach: Differentiable Augmentation



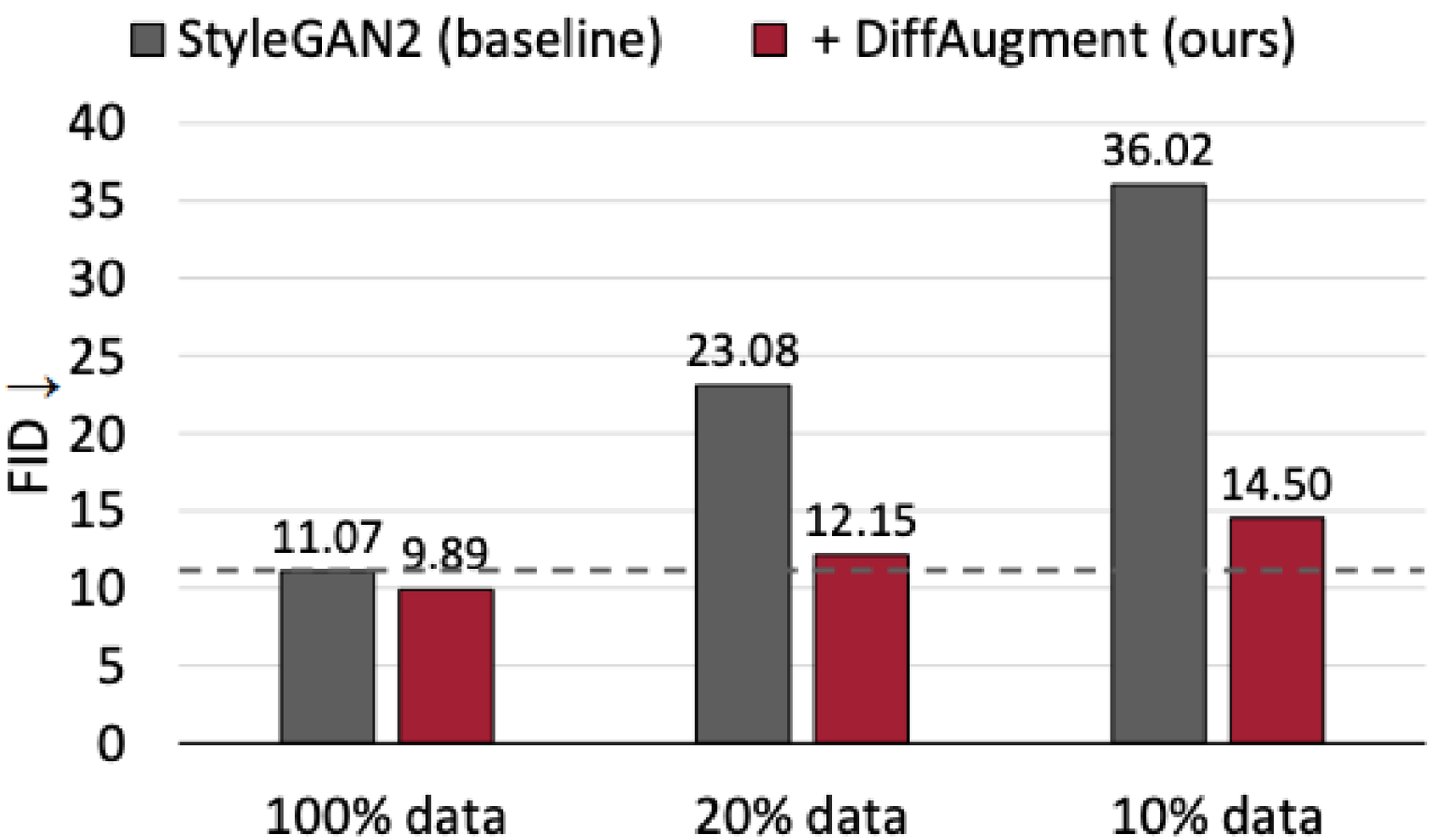




Our approach (DiffAugment): Augment reals + fakes for both $\boldsymbol{D}$ and $\boldsymbol{G}$

Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

CIFAR-10 (unconditional GANs)



Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

ImageNet Generation (25% training data)

IS: 46.5 FID: 25.37

BigGAN (baseline)

IS: **74.2** FID: **13.28**

+ DiffAugment (ours)

Low-Shot Generation

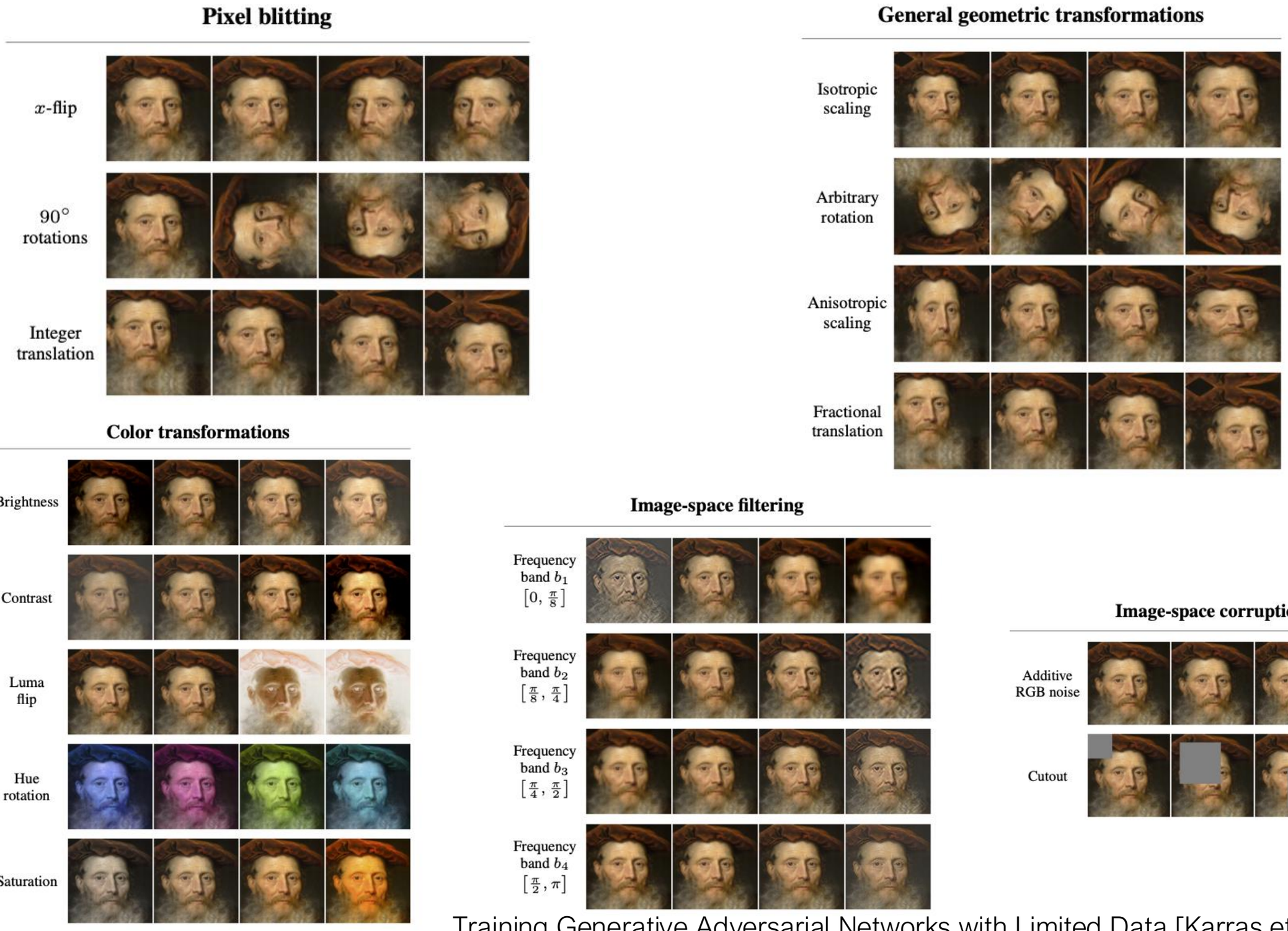
```python
from DiffAugment_pytorch import DiffAugment
# from DiffAugment_tf import DiffAugment
policy = 'color,translation,cutout' # If your dataset is as small as ours (e.g.,
# hundreds of images), we recommend using the strongest Color + Translation + Cutout.
# For large datasets, try using a subset of transformations in ['color', 'translation', 'cutout'].
# Welcome to discover more DiffAugment transformations!

...
# Training loop: update D
reals = sample_real_images() # a batch of real images
z = sample_latent_vectors()
fakes = Generator(z) # a batch of fake images
real_scores = Discriminator(DiffAugment(reals, policy=policy))
fake_scores = Discriminator(DiffAugment(fakes, policy=policy))
# Calculating D's loss based on real_scores and fake_scores...
...

...
# Training loop: update G
z = sample_latent_vectors()
fakes = Generator(z) # a batch of fake images
fake_scores = Discriminator(DiffAugment(fakes, policy=policy))
# Calculating G's loss based on fake_scores...
...
```

Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

StyleGAN2-ADA



Training Generative Adversarial Networks with Limited Data [Karras et al., NeurIPS 2020]

StyleGAN2-ADA

Adaptative data augmentation

$$r_t = \mathbb{E}[\mathbf{sign}(D_{\text{train}})]$$

$r_t = 0$ no overfitting, decrease augmentation

r=1 complete overfitting, increase augmentation

Other metrics to consider:

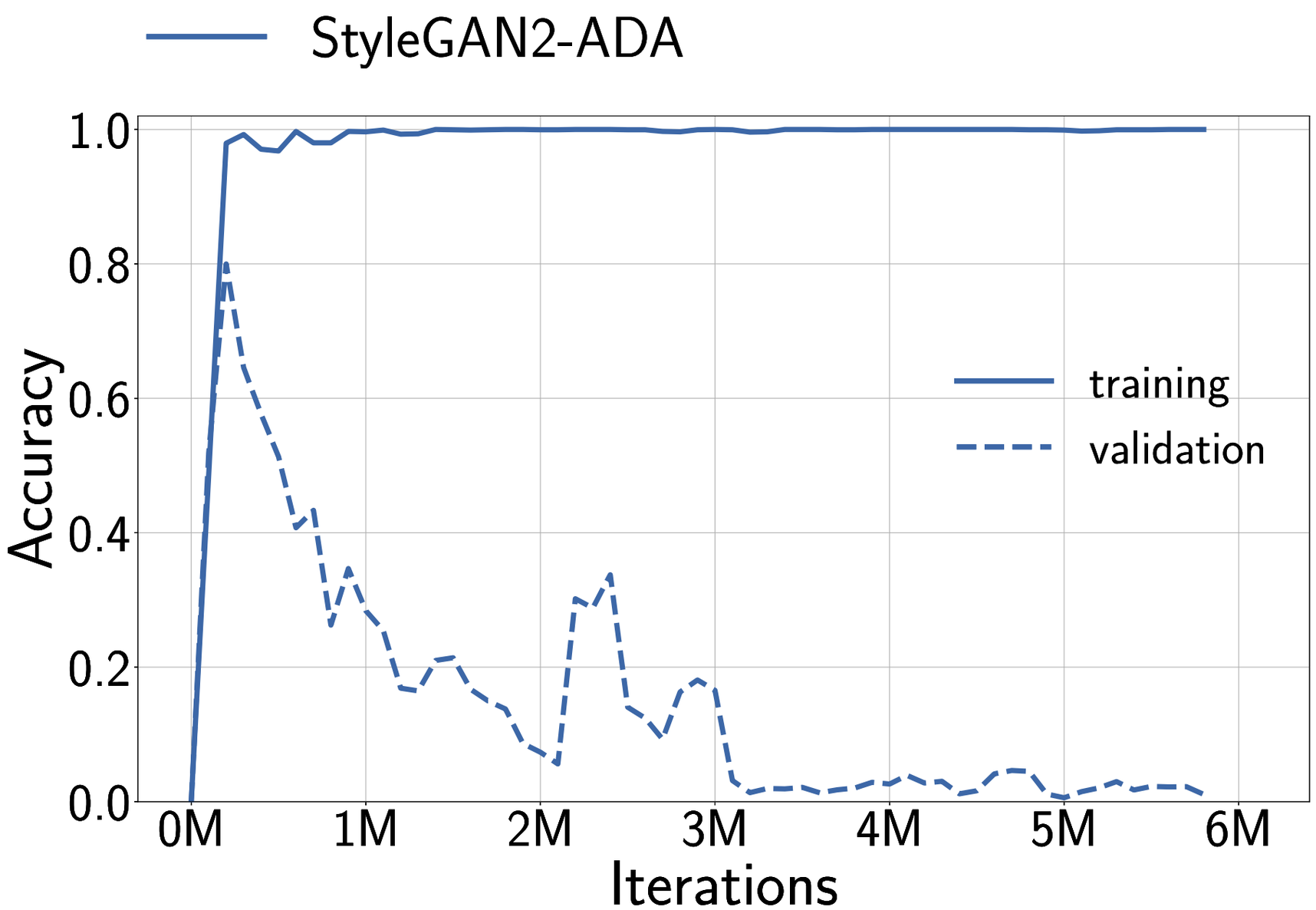
$$\frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]}$$

$$\mathbb{E}[D_{\text{train}}]$$

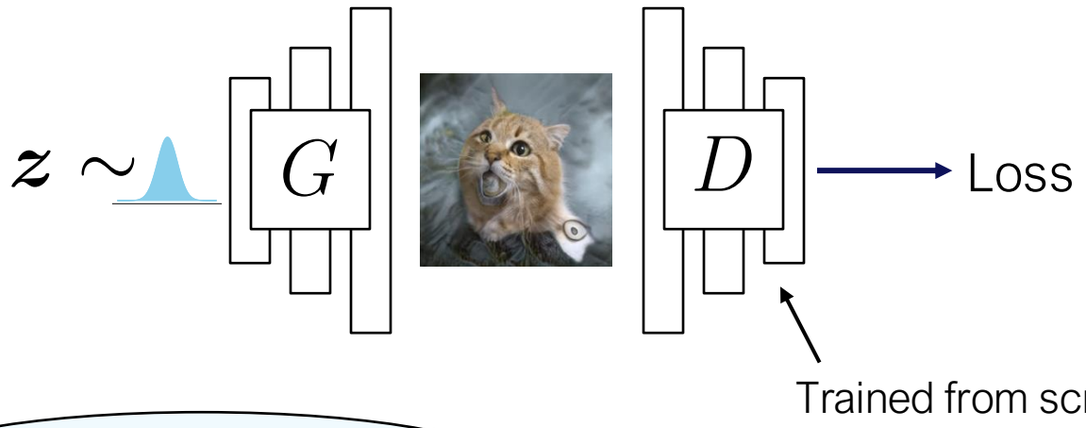
Training Generative Adversarial Networks with Limited Data [Karras et al., NeurIPS 2020]

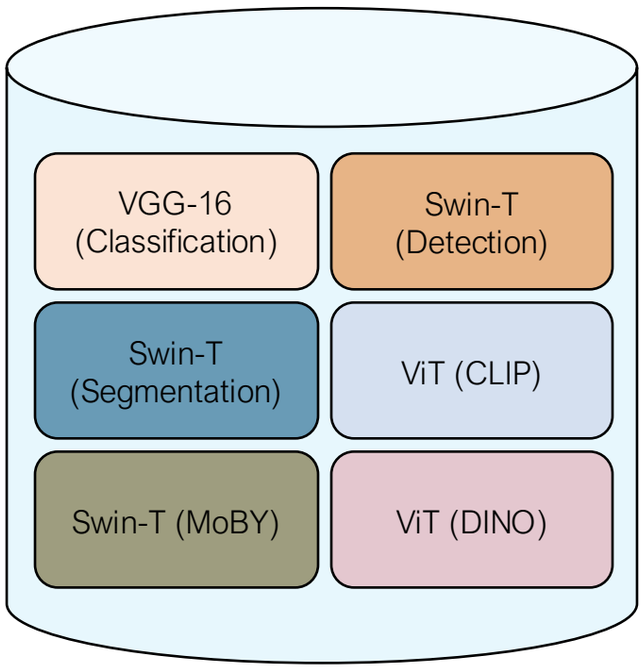
Training methods

Discriminator is still Overfitting



Standard GAN training



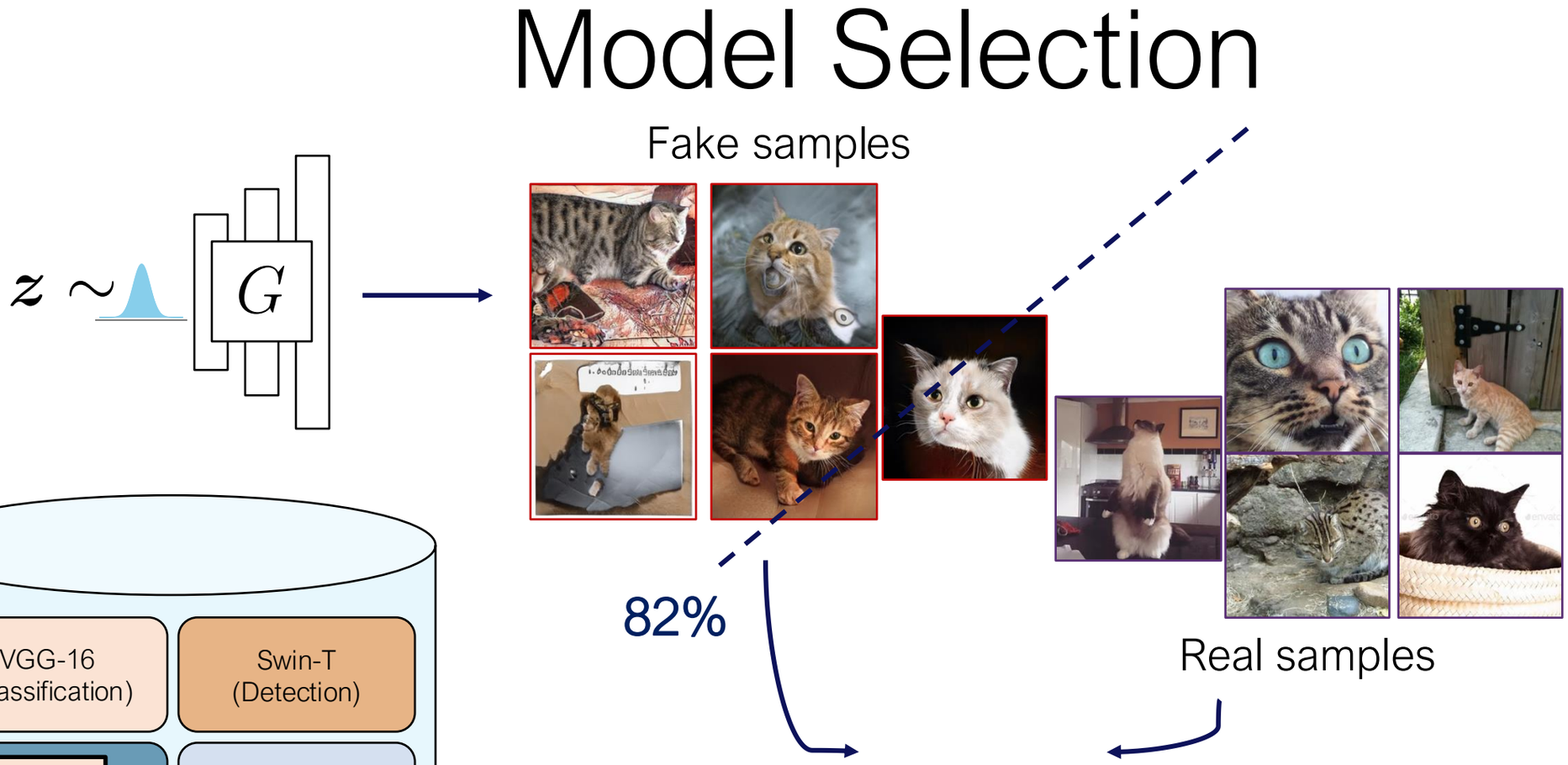


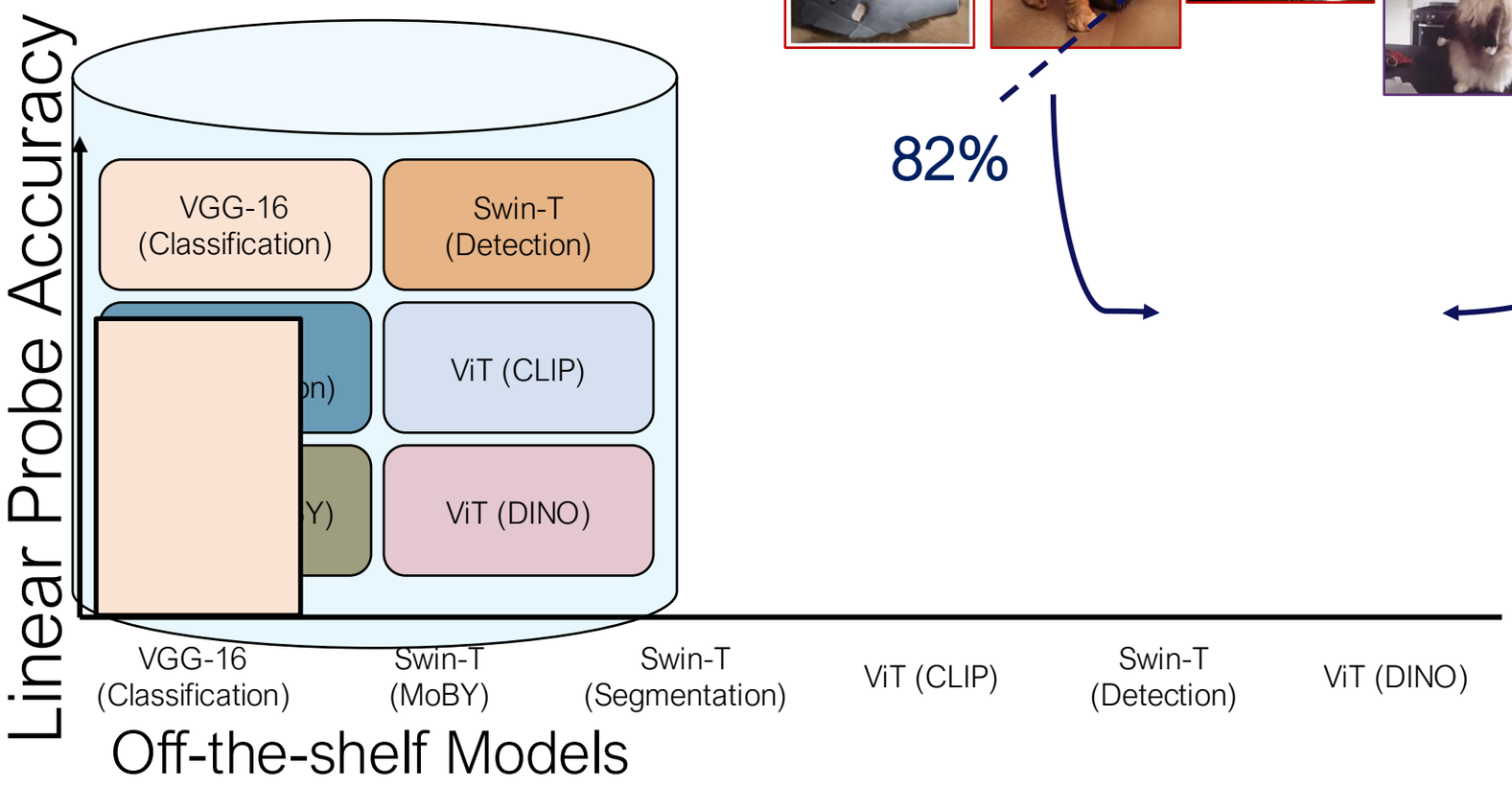
Off-the-shelf Models

Which pretrained models to use?

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

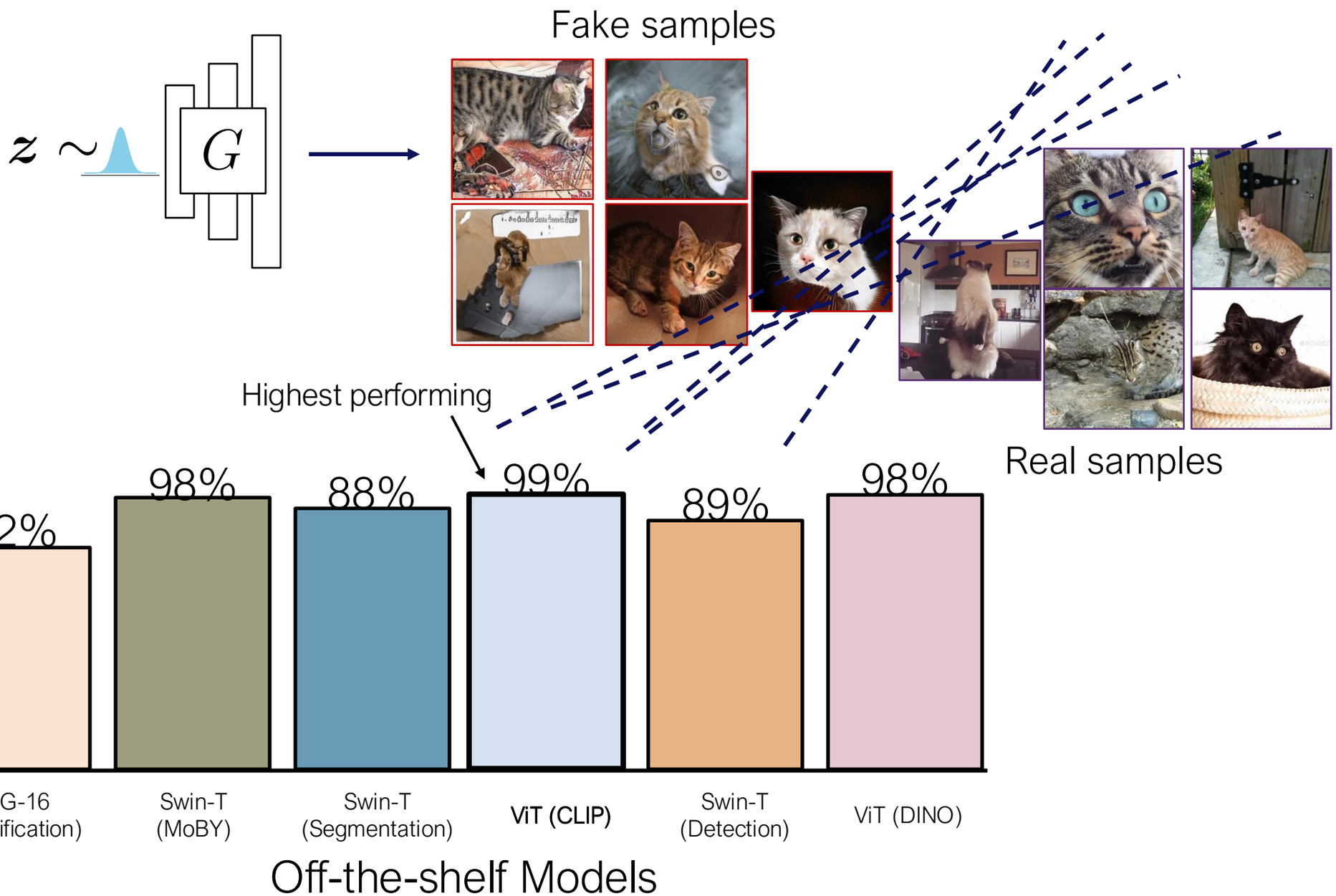
Model Selection





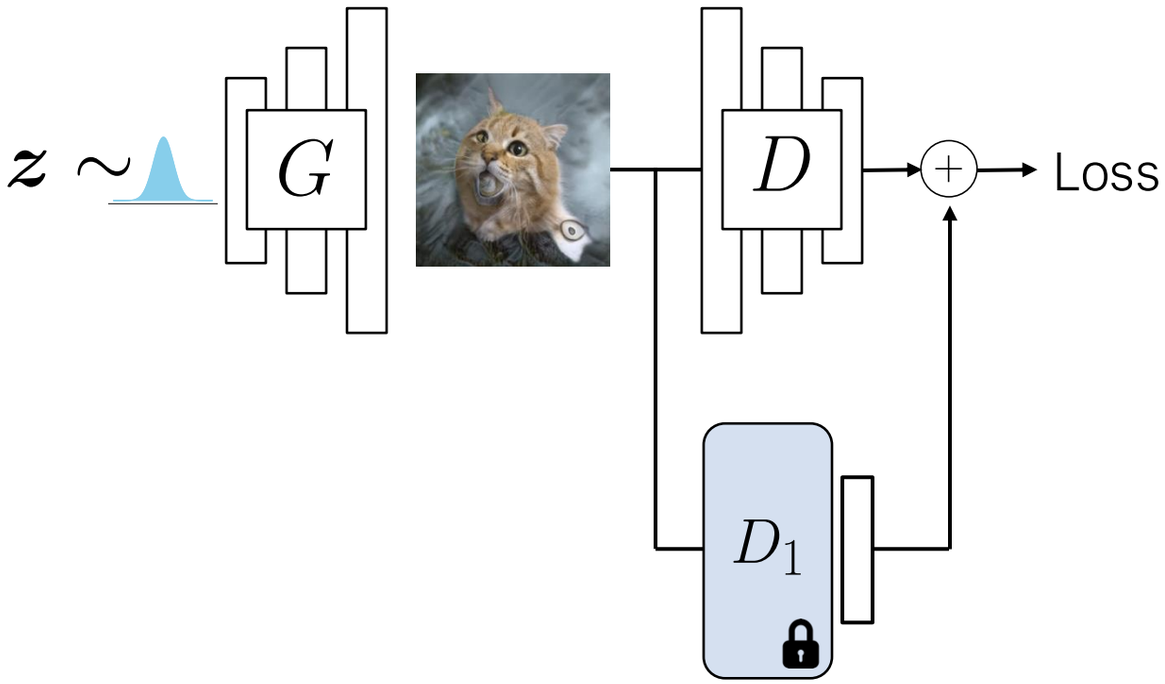
Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Model Selection



Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Vision-aided GAN training

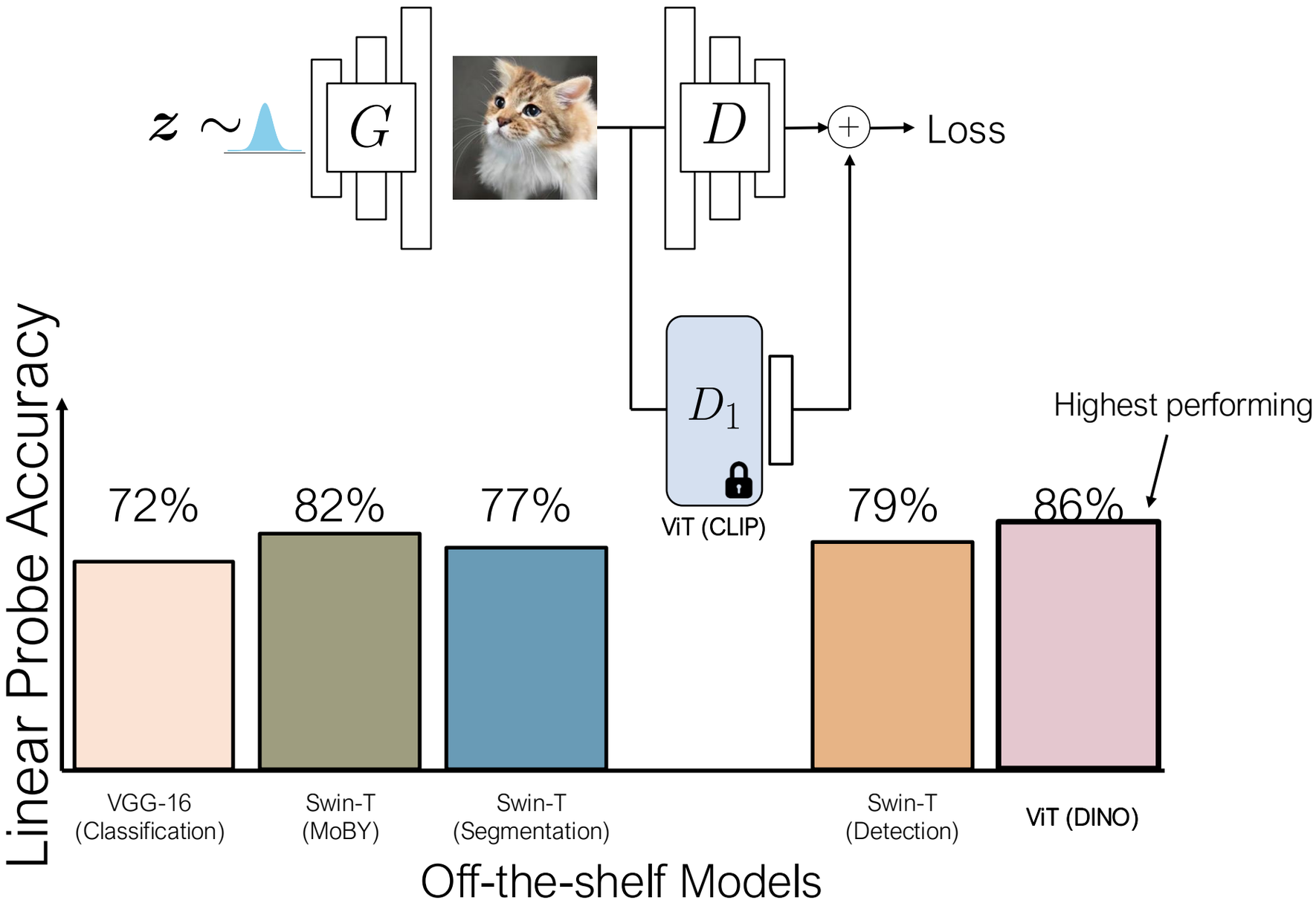


VGG-16 (Classification) | Swin-T (MoBY) | Swin-T (Segmentation) | **ViT (CLIP)** | Swin-T (Detection) | ViT (DINO)

Off-the-shelf Models

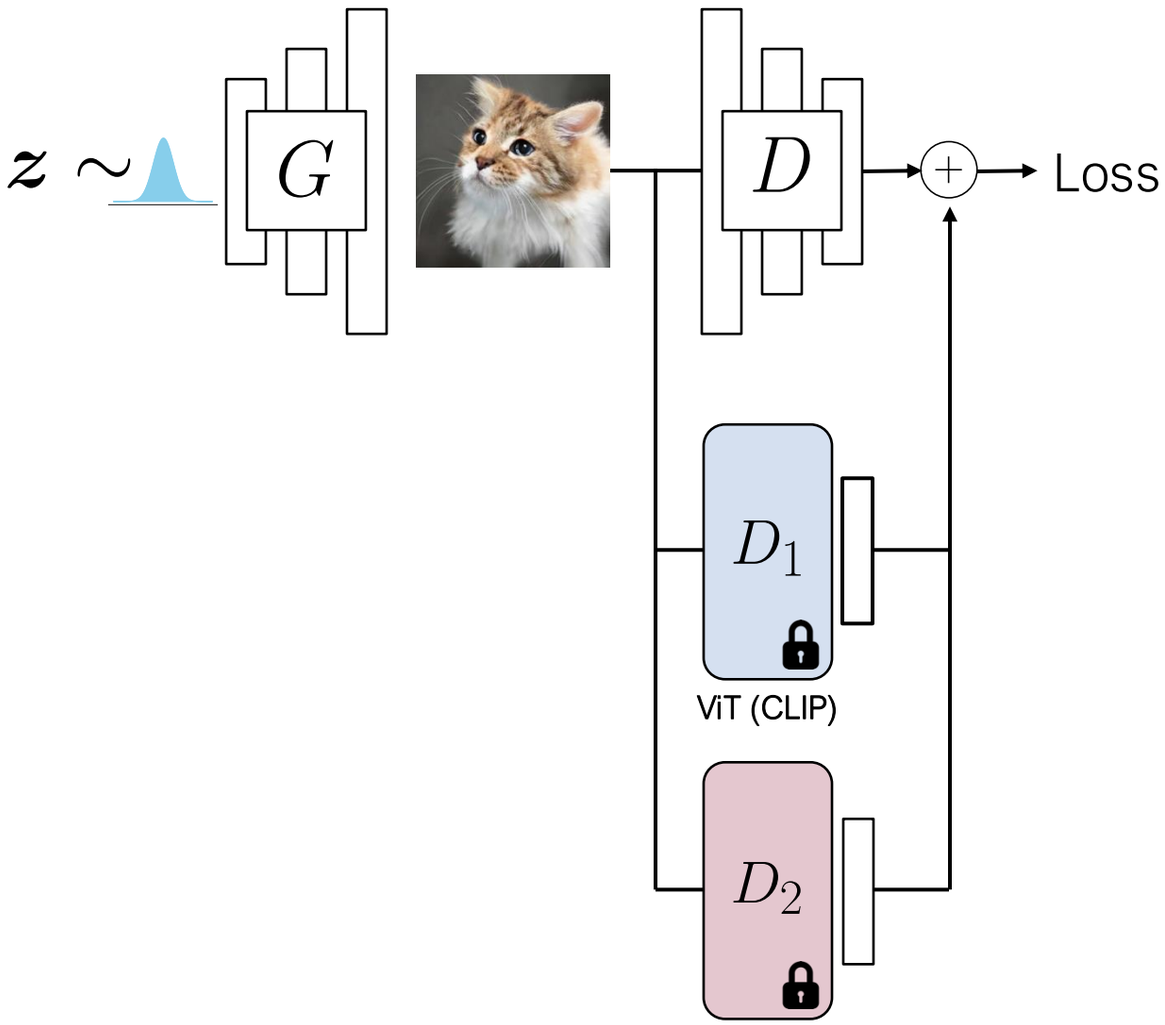
Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Add 2nd Vision-aided discriminator



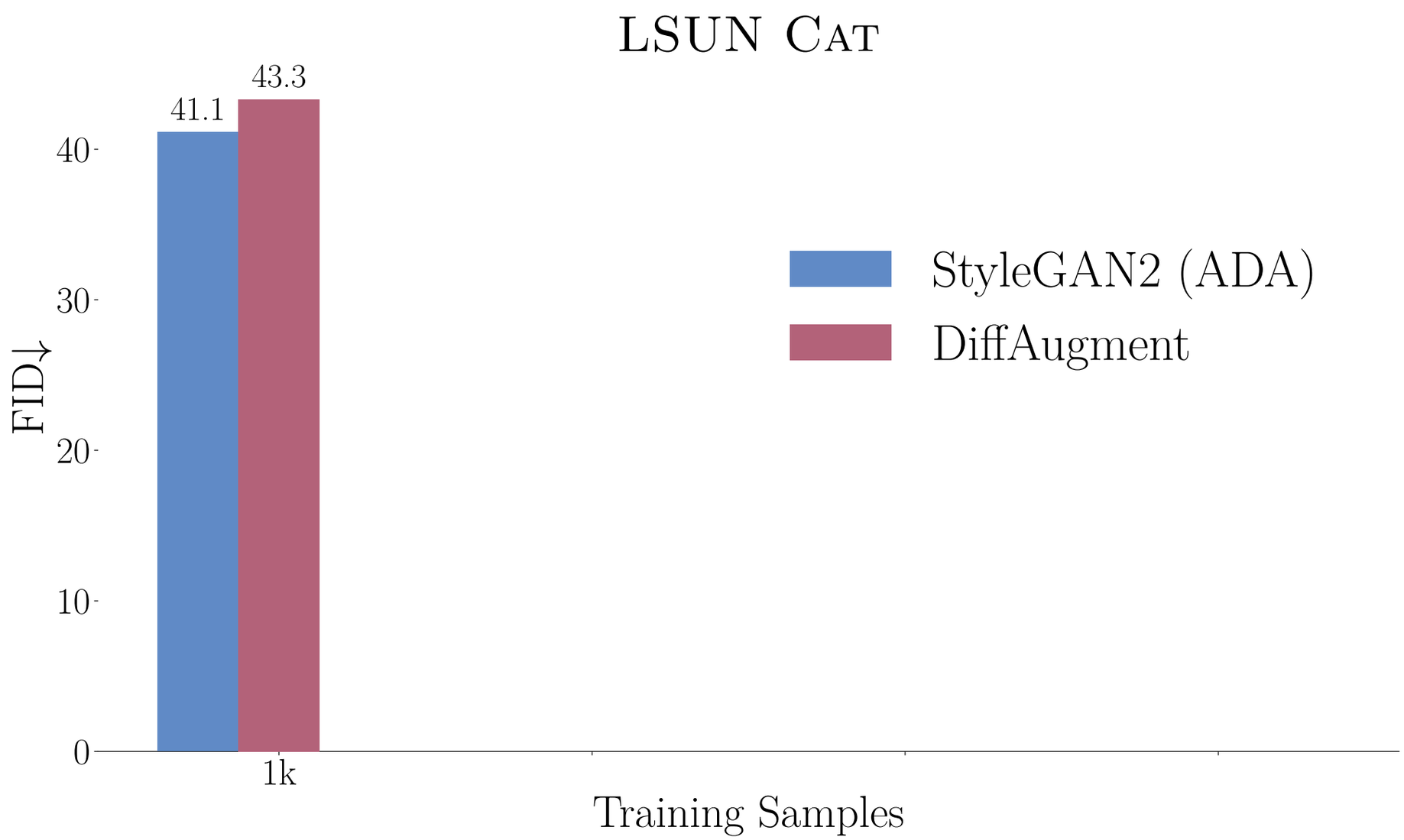
Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Add 2nd Vision-aided discriminator

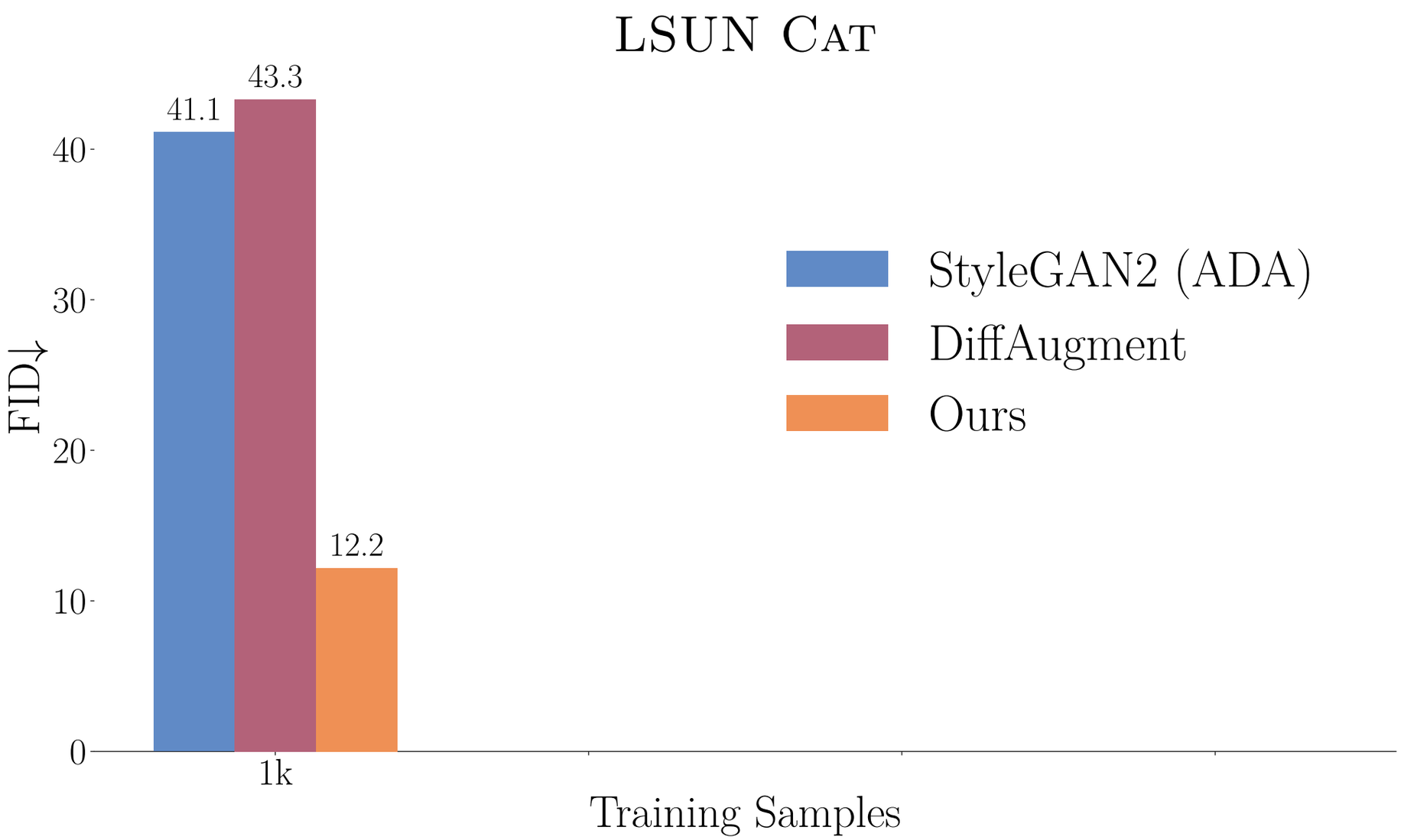


Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

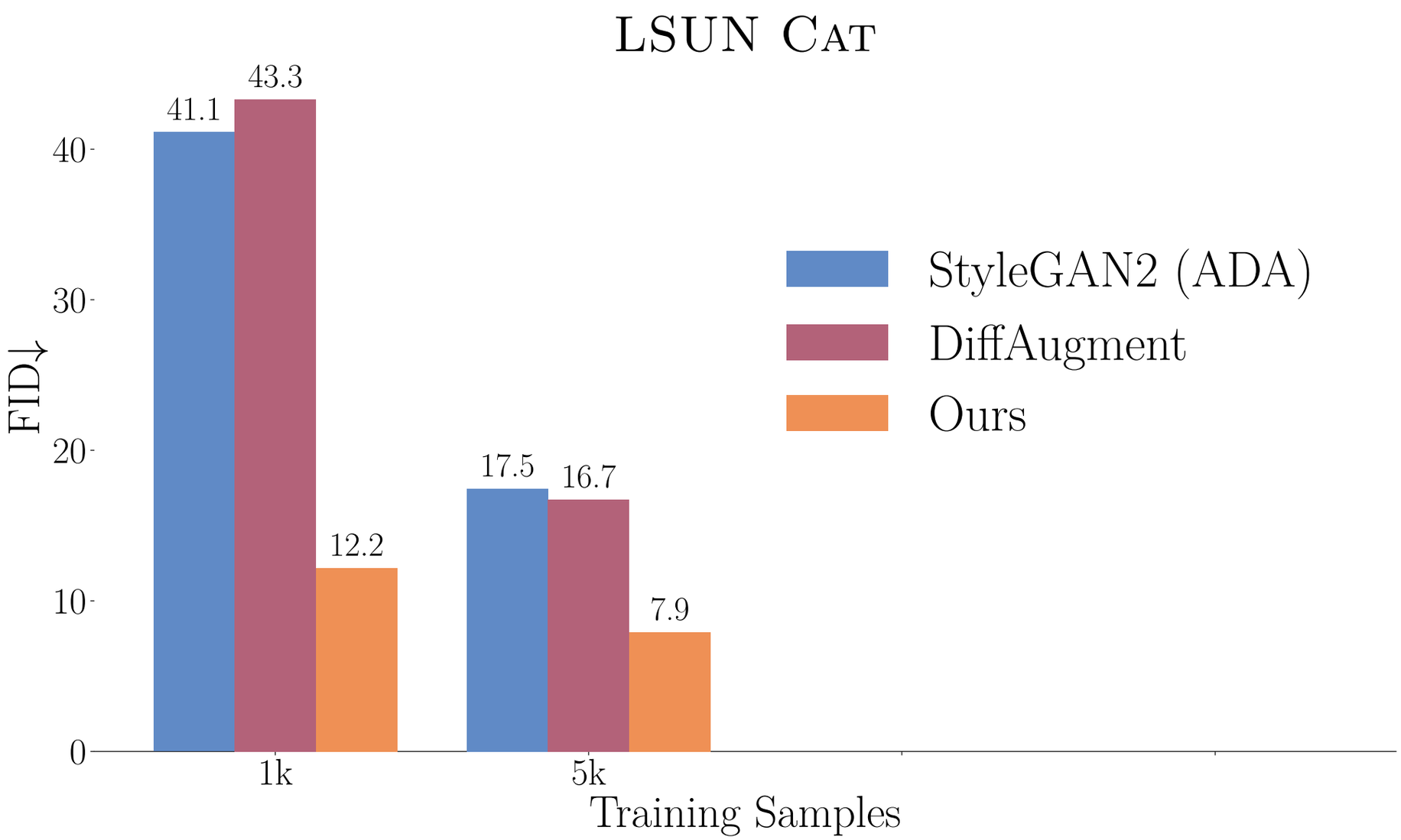
Benefit with varying training samples



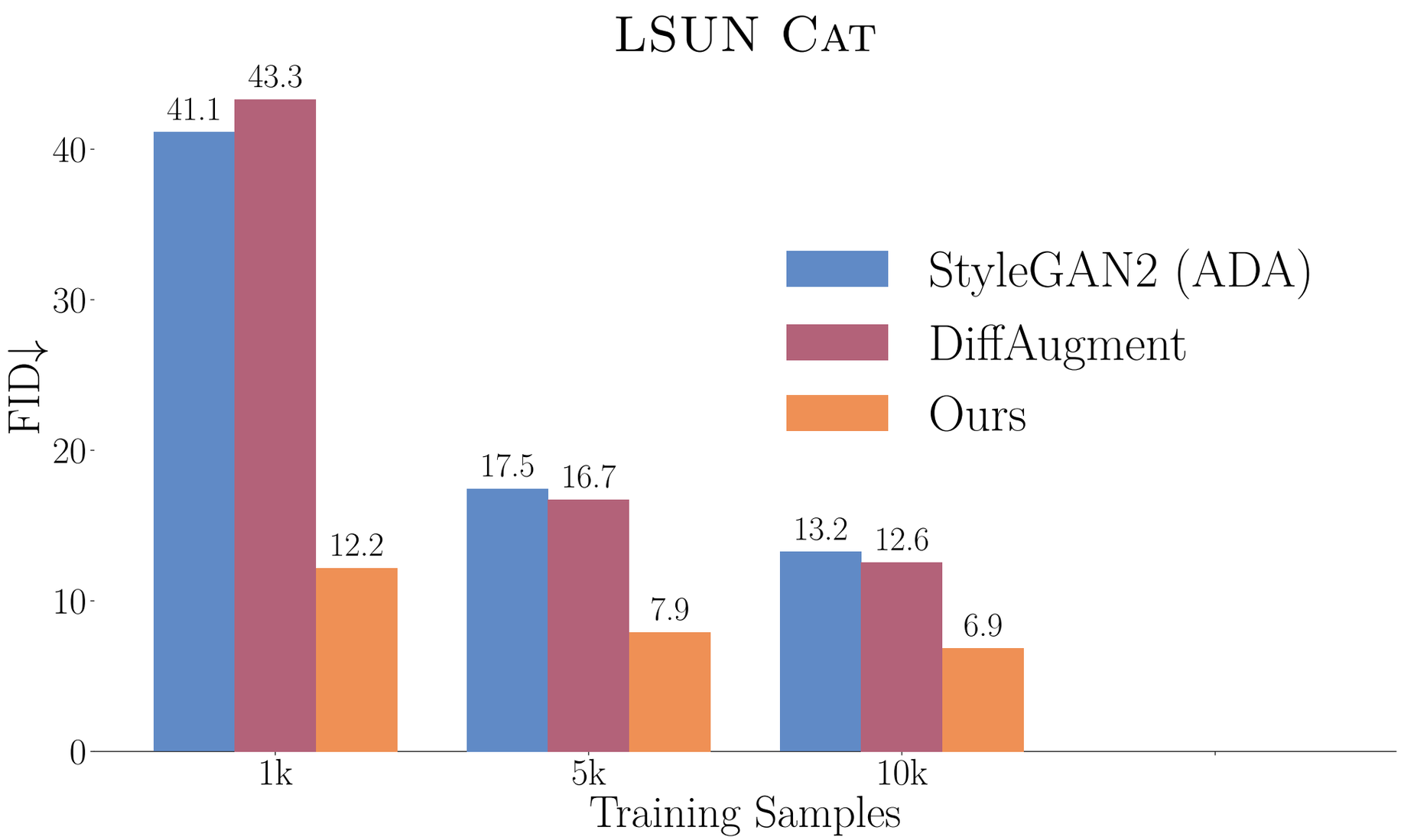
Benefit with varying training samples



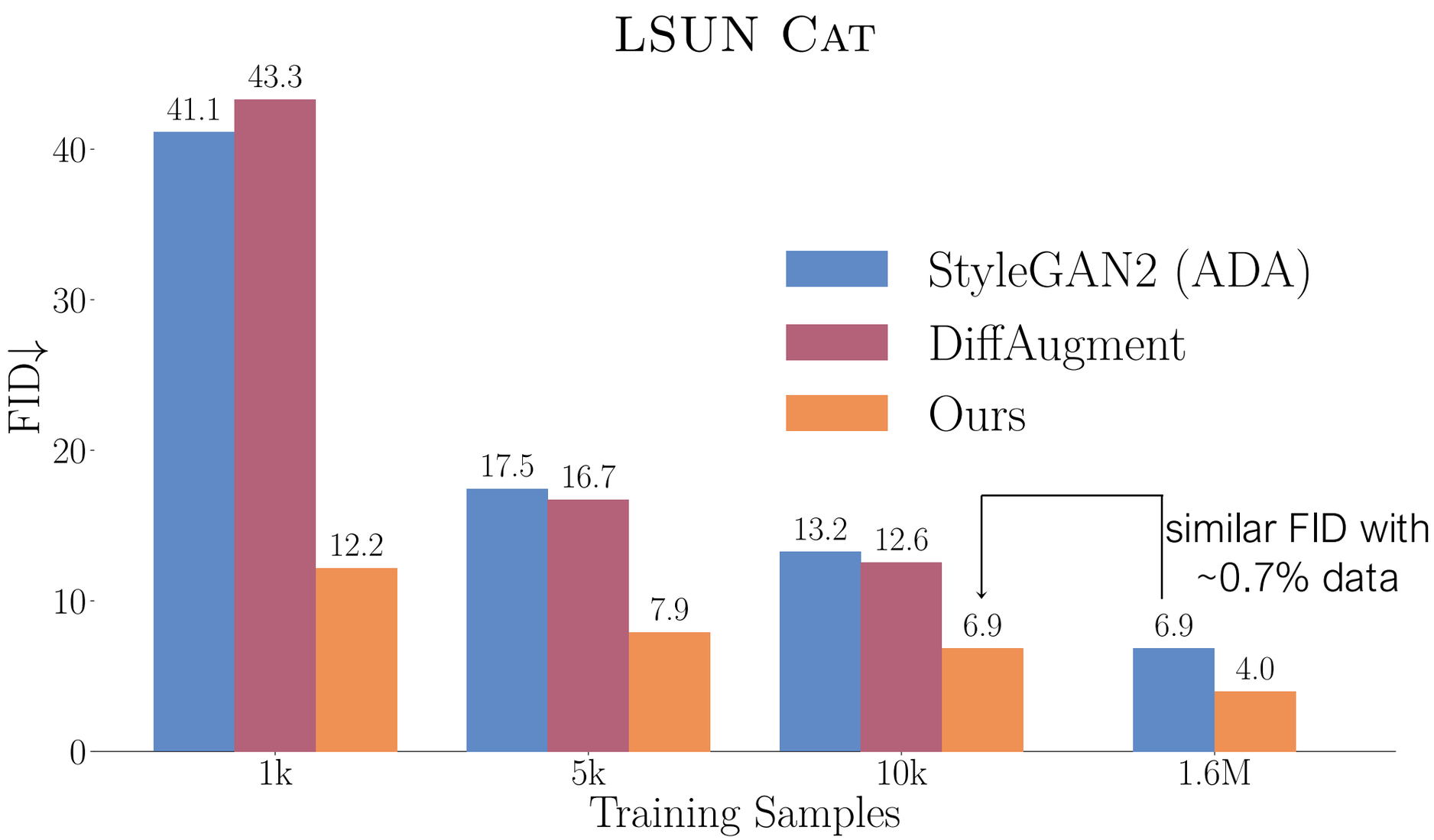
Benefit with varying training samples



Benefit with varying training samples

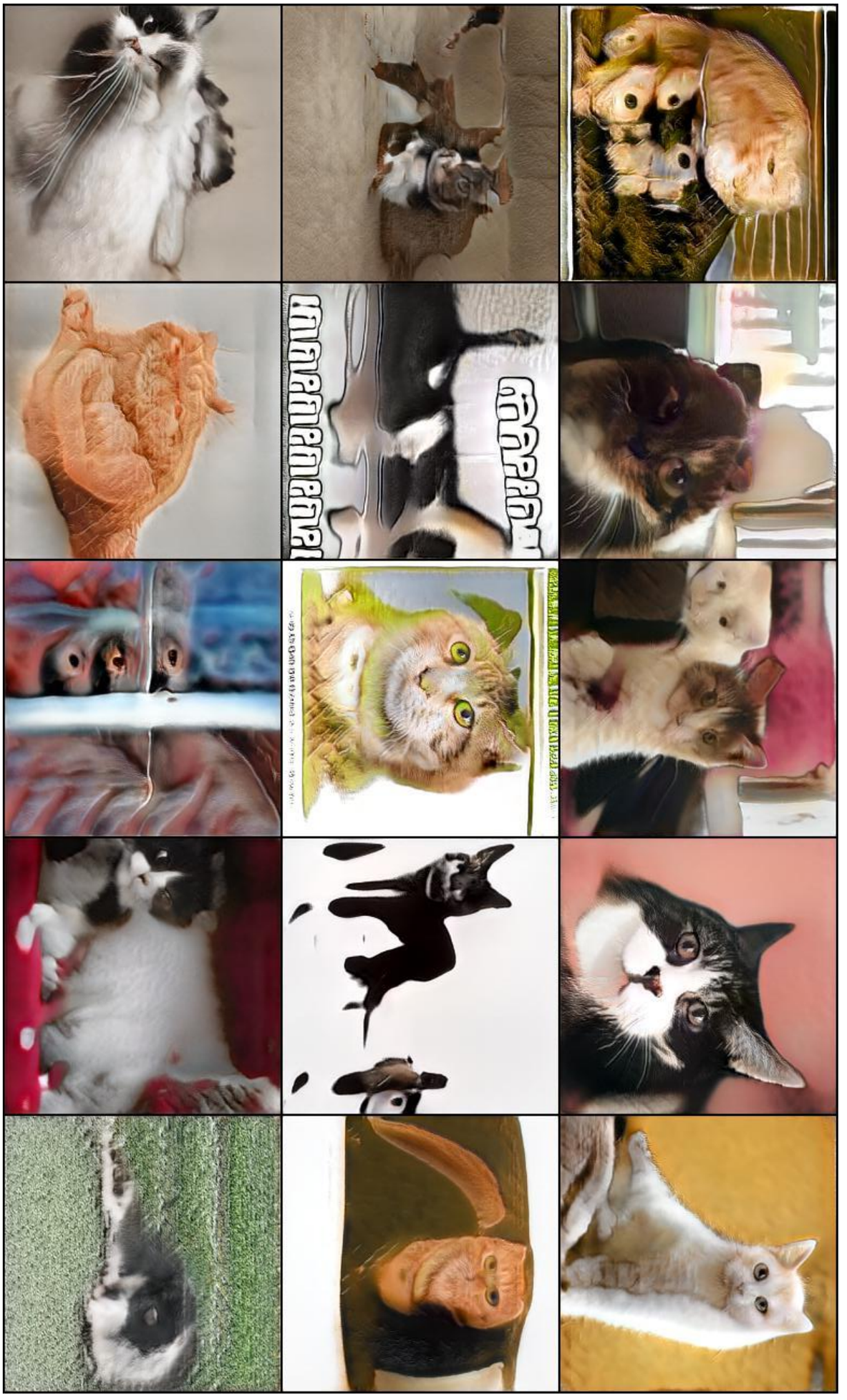


Benefit with varying training samples



StyleGAN2-ADA

LSUN Cat 1k



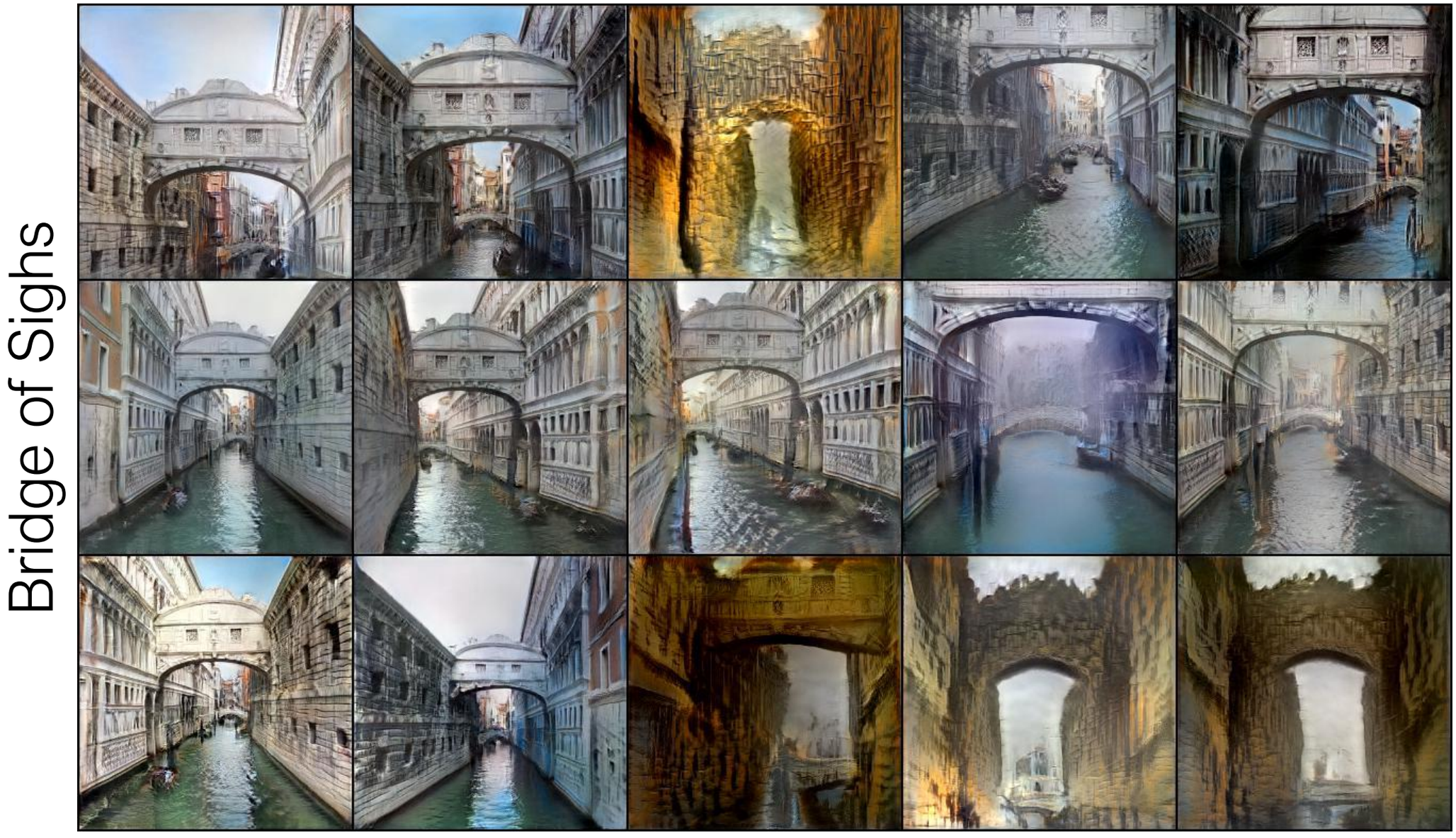
Improved Samples

Improved Samples

Ours

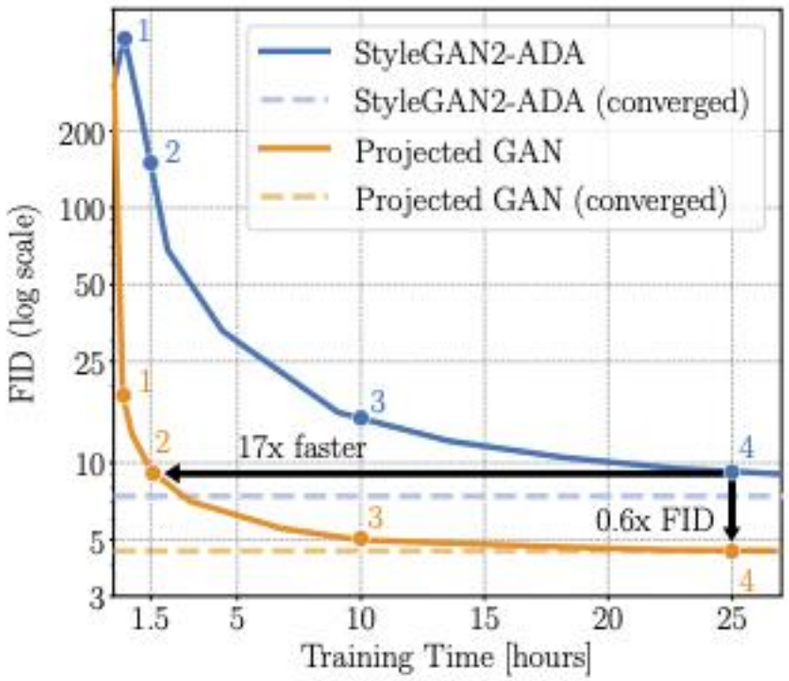
LSUN CAT 1k

Low-shot Generation with 100 samples

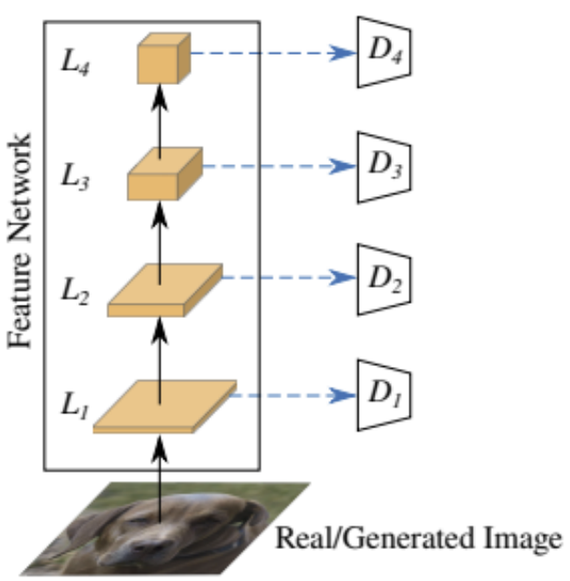


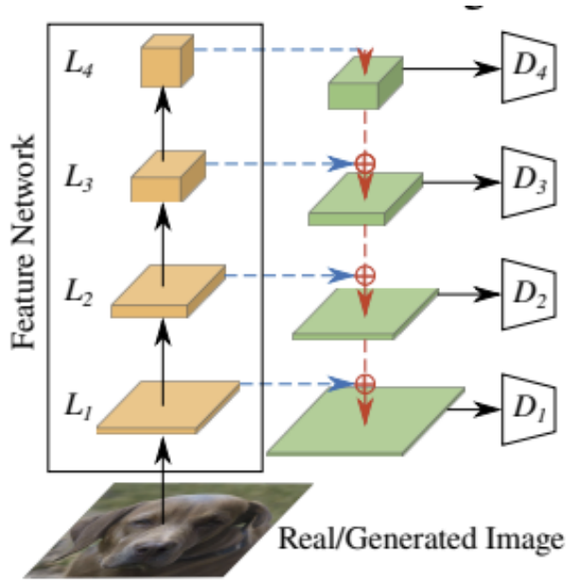
Low-shot Generation with 100 samples

Faster Convergence with Projected GANs



Dashed blue arrows :
1x1 conv
with random weights





Dashed red arrows:
3x3 conv
with random weights

use a FastGAN generator (https://arxiv.org/abs/2101.04775)

Projected GANs Converge Faster. Sauer et al., NeurIPS 2021

Combining Perceptual Loss and GAN Loss

Idea 1: add them together (many papers did that. It works)

Idea 2: Pre-trained features + trainable MLP layers
= Perceptual Discriminator

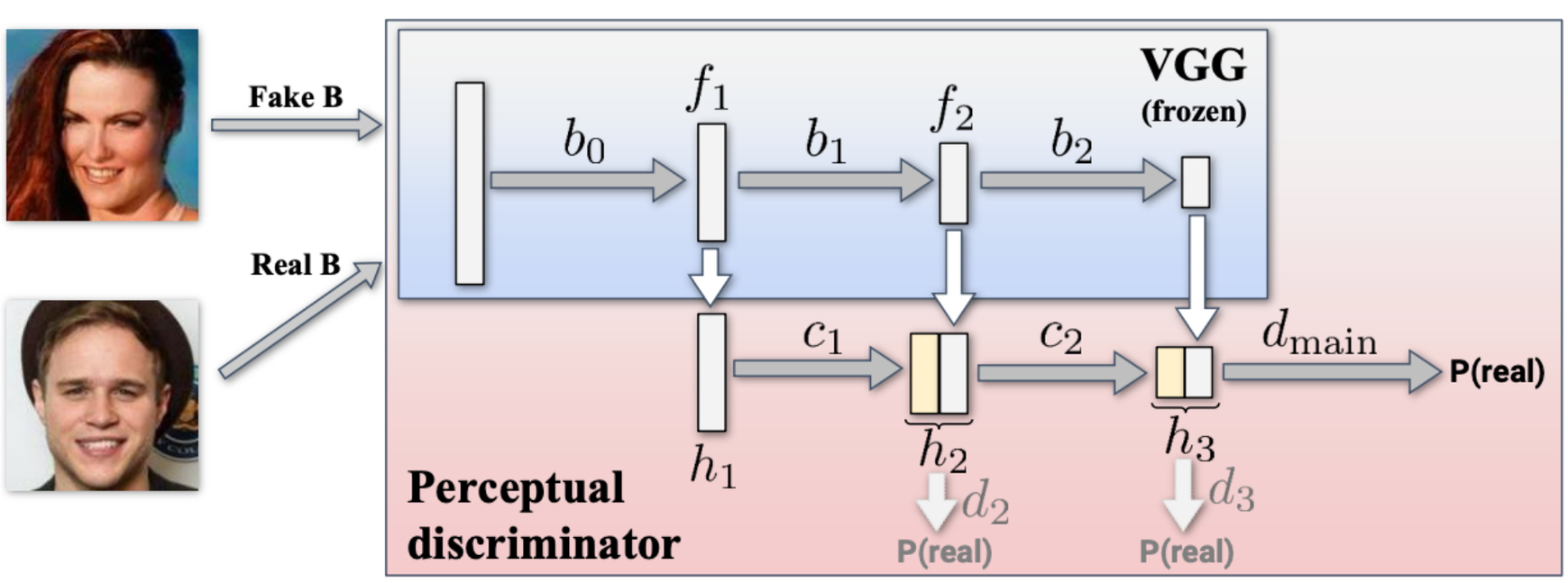


Image Manipulation with Perceptual Discriminators [Sungatullina et al. ECCV 2018]
Using multiple pre-trained models: Vision-aided GANs [Kumari et al., 2021]
Using random projection head: Projected GANs [Sauer et al., NeurIPS 2021]
Conditional discriminator: Enhancing photorealism enhancement [Richter et al., 2020]

Large-scale GAN training

Recent trends in text-to-image synthesis

GAN-based models

Single-step inference

- GAN-CLS (ICML 2016)
- StackGAN (ICCV 2017)
- AttnGAN (CVPR 2018)
- DM-GAN (CVPR 2019)
- XMC-GAN (CVPR 2021)
- LAFITE (CVPR 2022)

A city street line with brick buildings and trees.

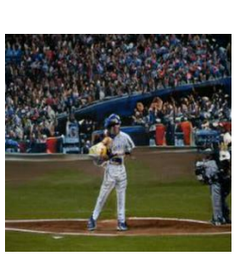
A street scene with a double-decker bus on the side of the road.

A group of young people riding snow boards down a snow covered hillside.

A great shot of a full kitchen and partially a table.

A baseball player taking a swing at an incoming ball.

A close up of a plate of broccoli and sauce.



A crowd watching baseball players at a game.

A grand building is topped with tower clocks and sits within a clear blue sky.

Images from LAFITE (Zhou et al., 2021)

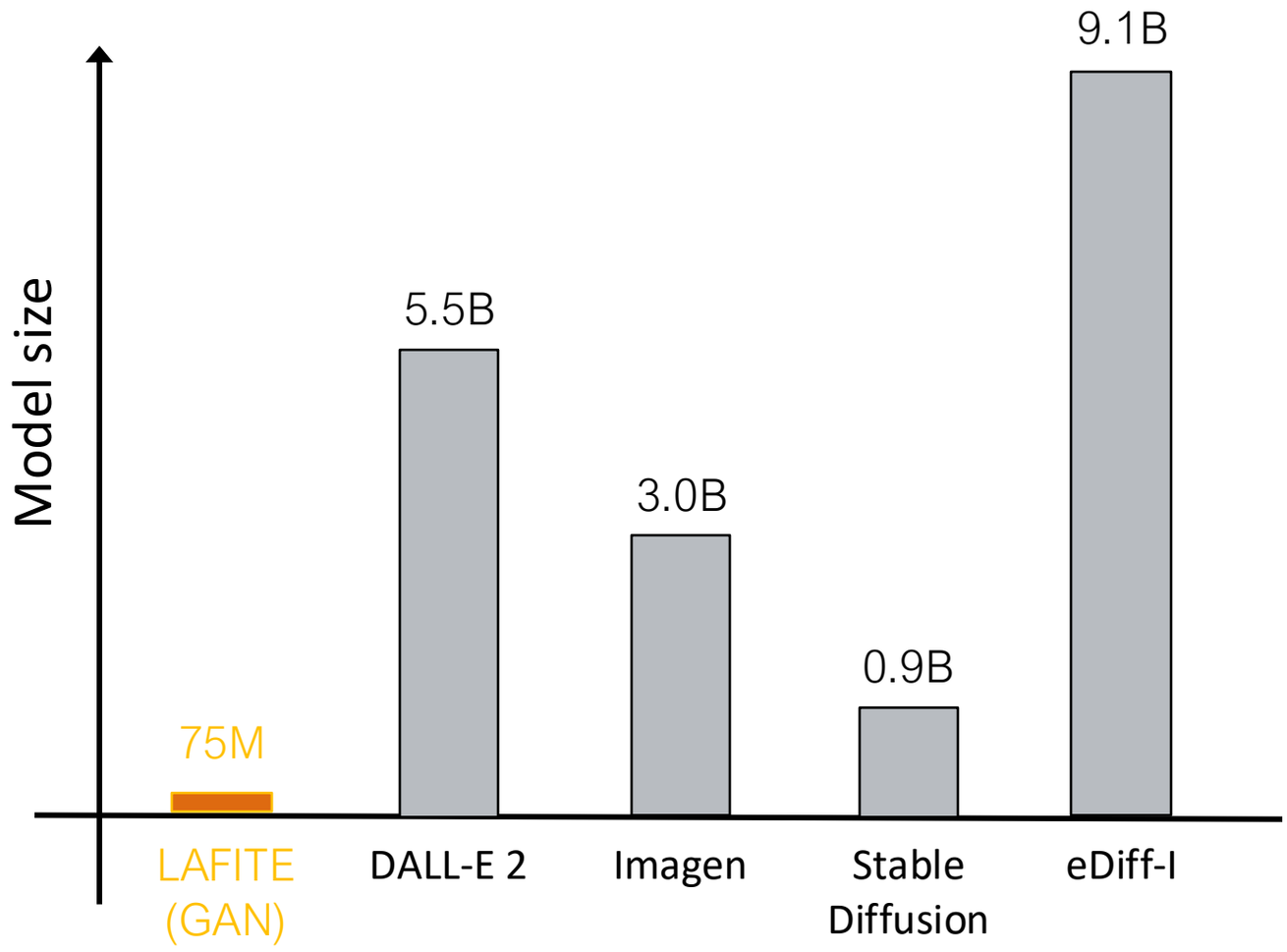
GAN was a popular choice for text-to-image synthesis, up until early 2022.

Recent trends in text-to-image synthesis

Diffusion/AR models

Multi-step inference

- GLIDE (ICML 2022)
- DALL-E 2 (arXiv, Apr. 2022)
- Imagen (NeurIPS 2022)
- Parti (TMLR 2022)
- Stable Diffusion (Aug. 2022)
- eDiff-I (arXiv , Nov. 2022)



A lot of research efforts *are now dedicated to diffusion models.*

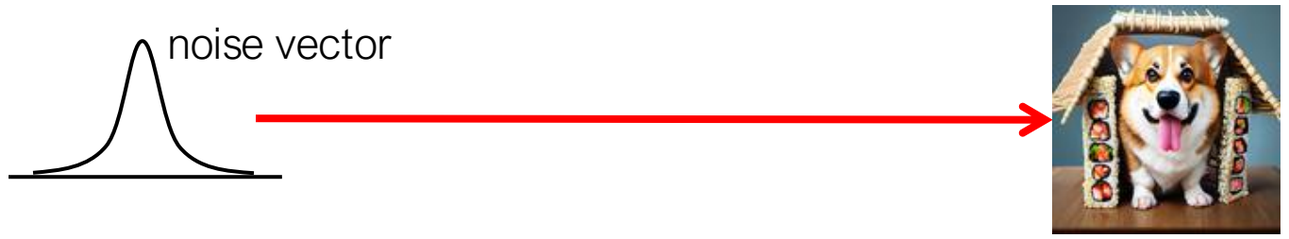
Recent trends in text-to-image synthesis

Diffusion/AR models

Multi-step inference

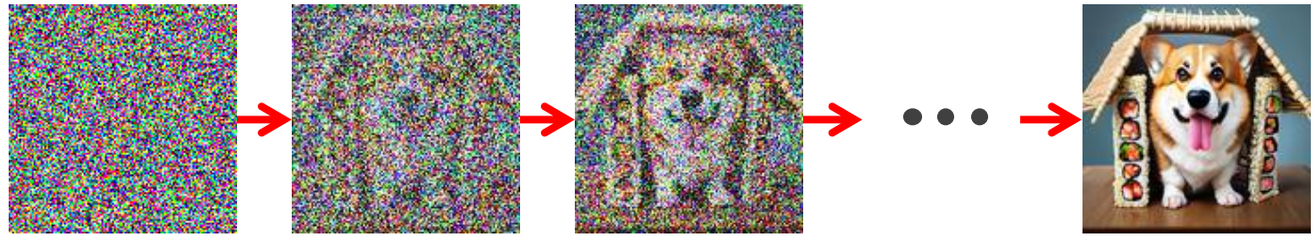
- GLIDE (ICML 2022)
- DALL-E 2 (arXiv, Apr. 2022)
- Imagen (NeurIPS 2022)
- Parti (TMLR 2022)
- Stable Diffusion (Aug. 2022)
- eDiff-I (arXiv , Nov. 2022)

GANs only need a single forward pass for generation.



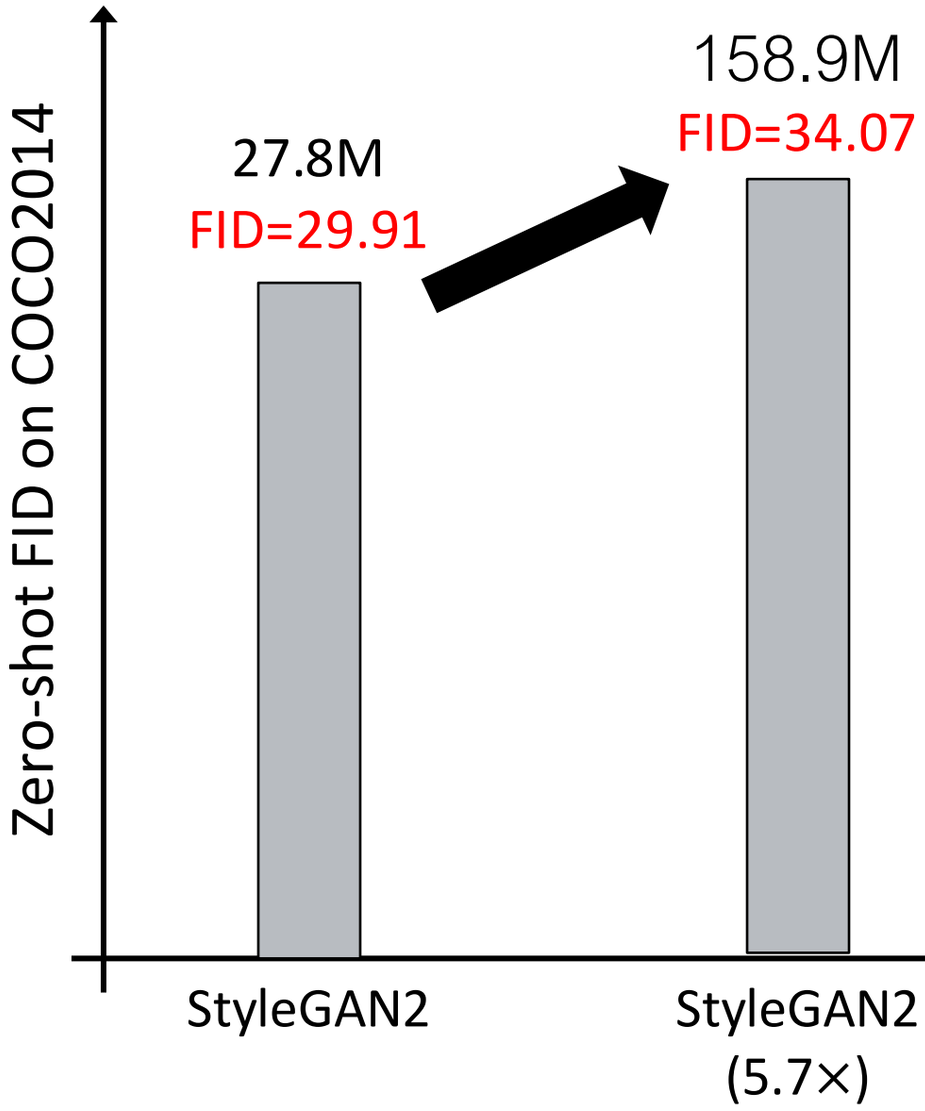
→ Fast, but low-quality

Diffusion methods require an iterative denoising process.



→ Slow, but high-quality

How we scale up GANs



We introduce and use:

1) *Self- and Cross-attention*
2) *Adaptive kernel selection*
3) *Multi-scale generator & discriminator*
4) *Matching-aware loss*
5) *CLIP contrastive loss*
6) *Vision-aided adversarial loss* [1]

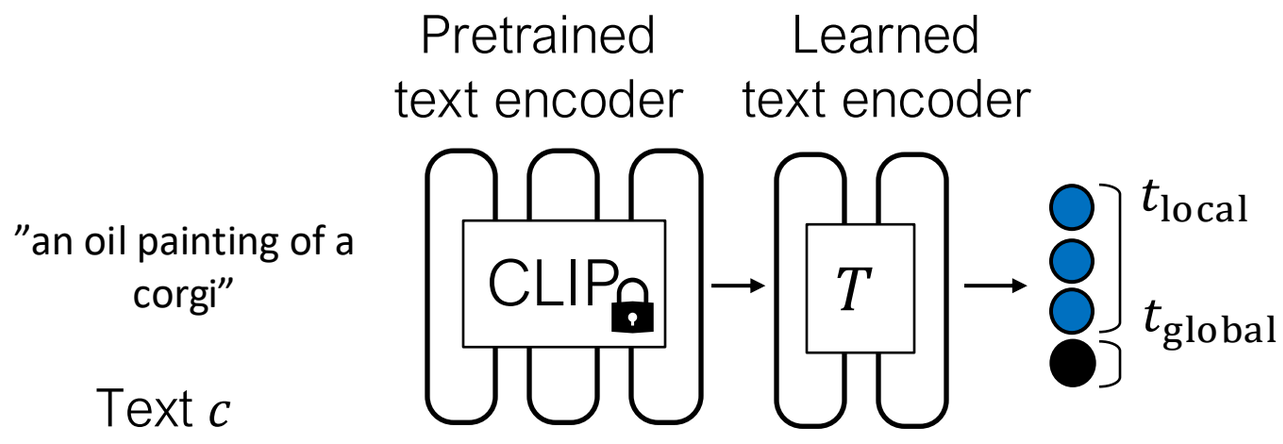
[1] Kumari et al., Ensembling Off-the-shelf Models for GAN Training, CVPR, 2022

Self-attention

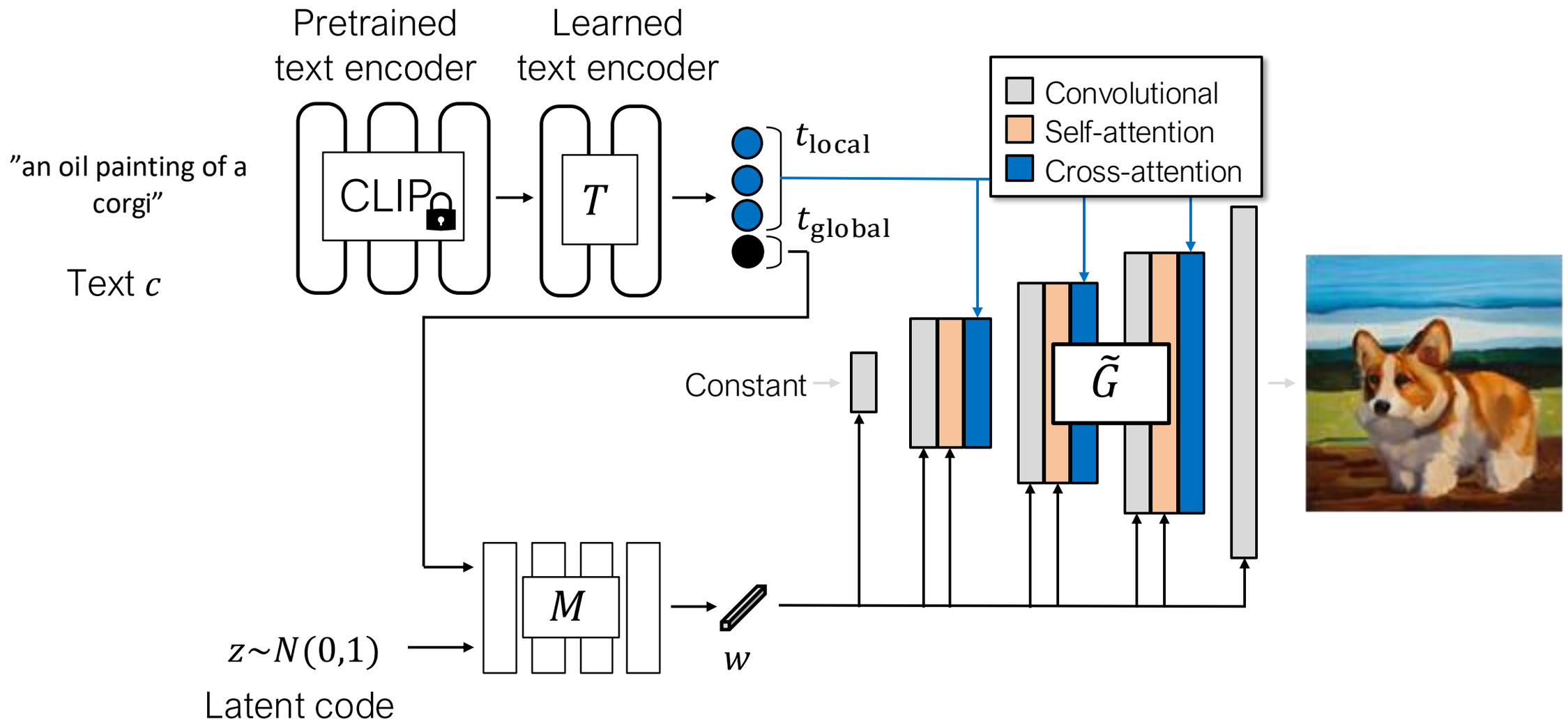
Results on ImageNet for class 0 (Tench)

Self-attention

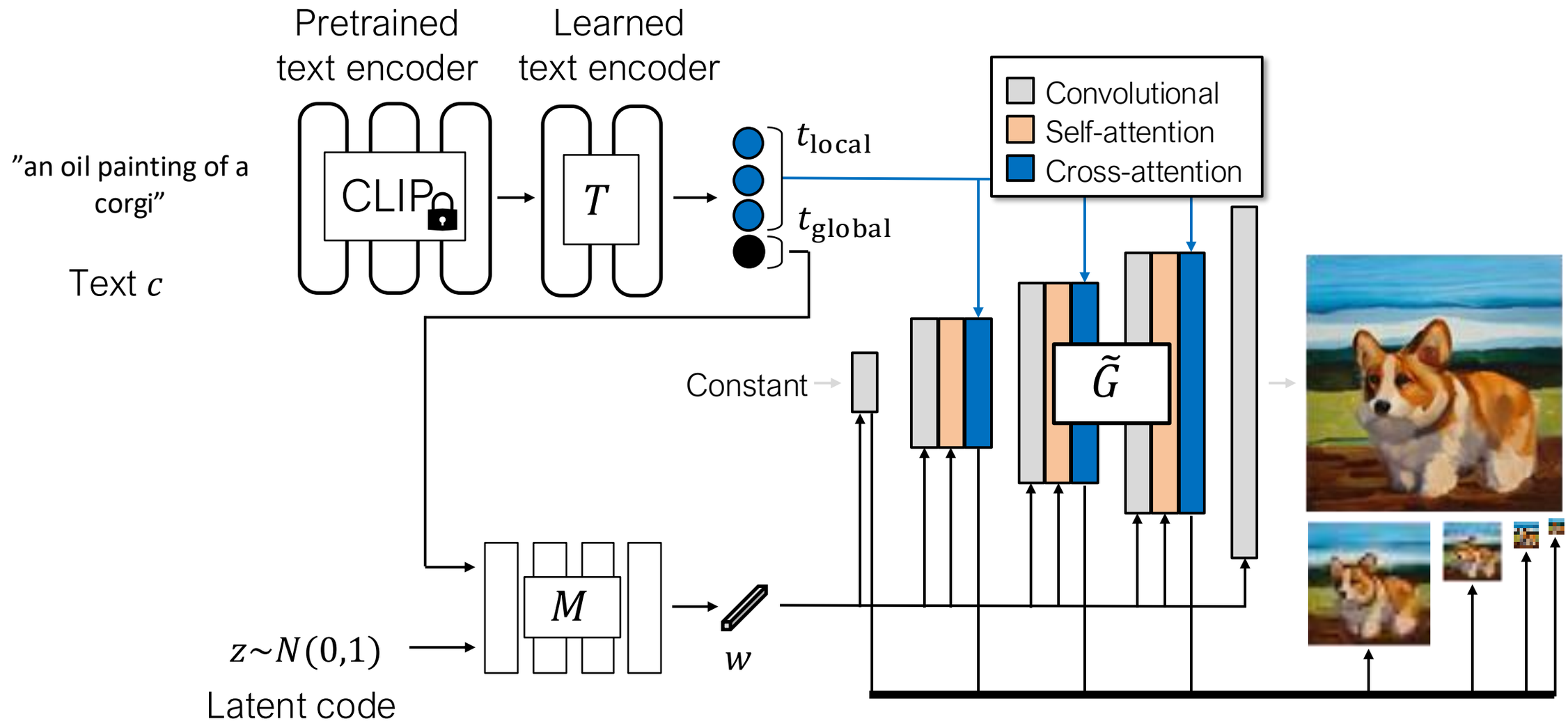
Multi-scale image generator



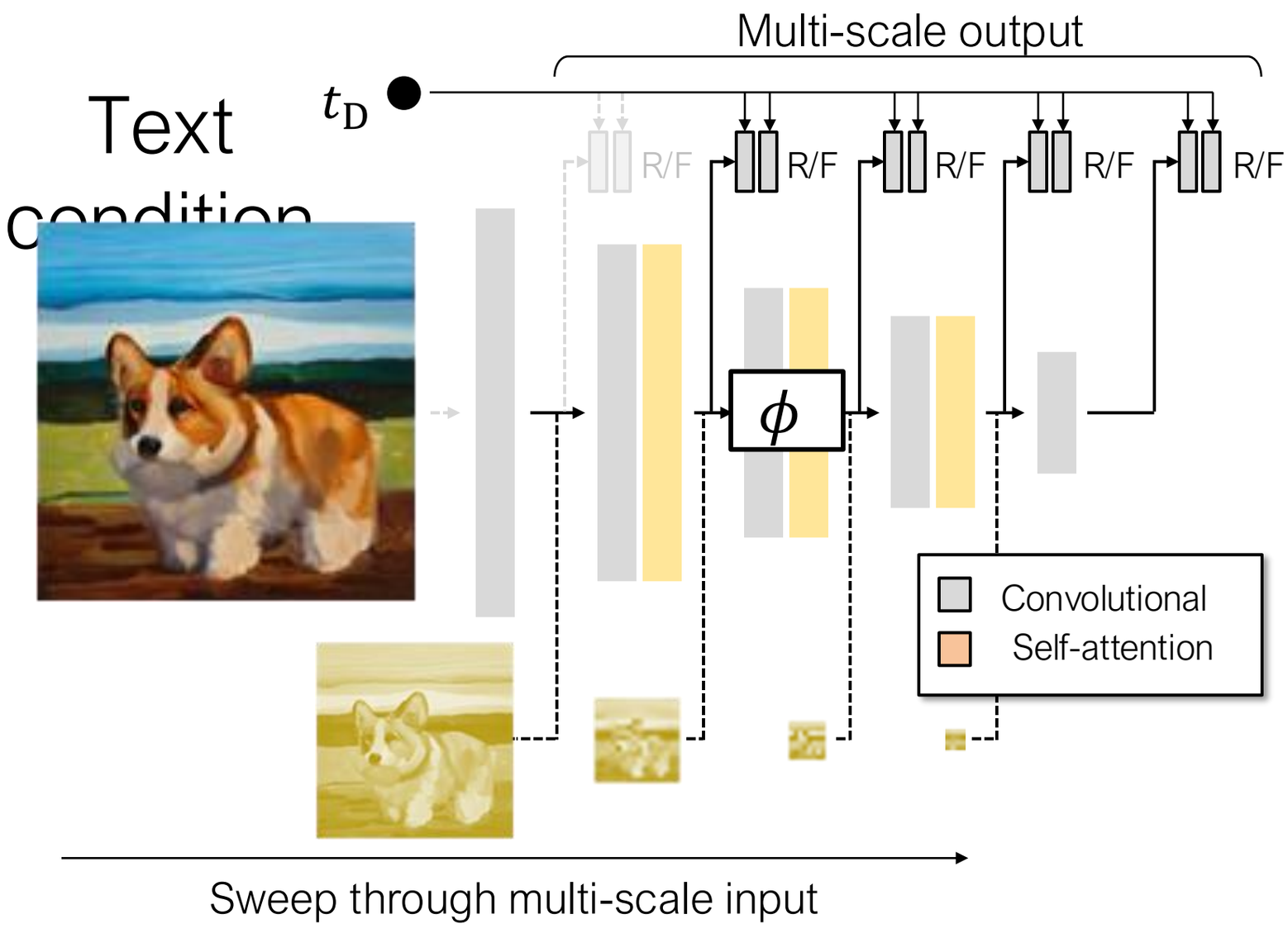
Multi-scale image generator



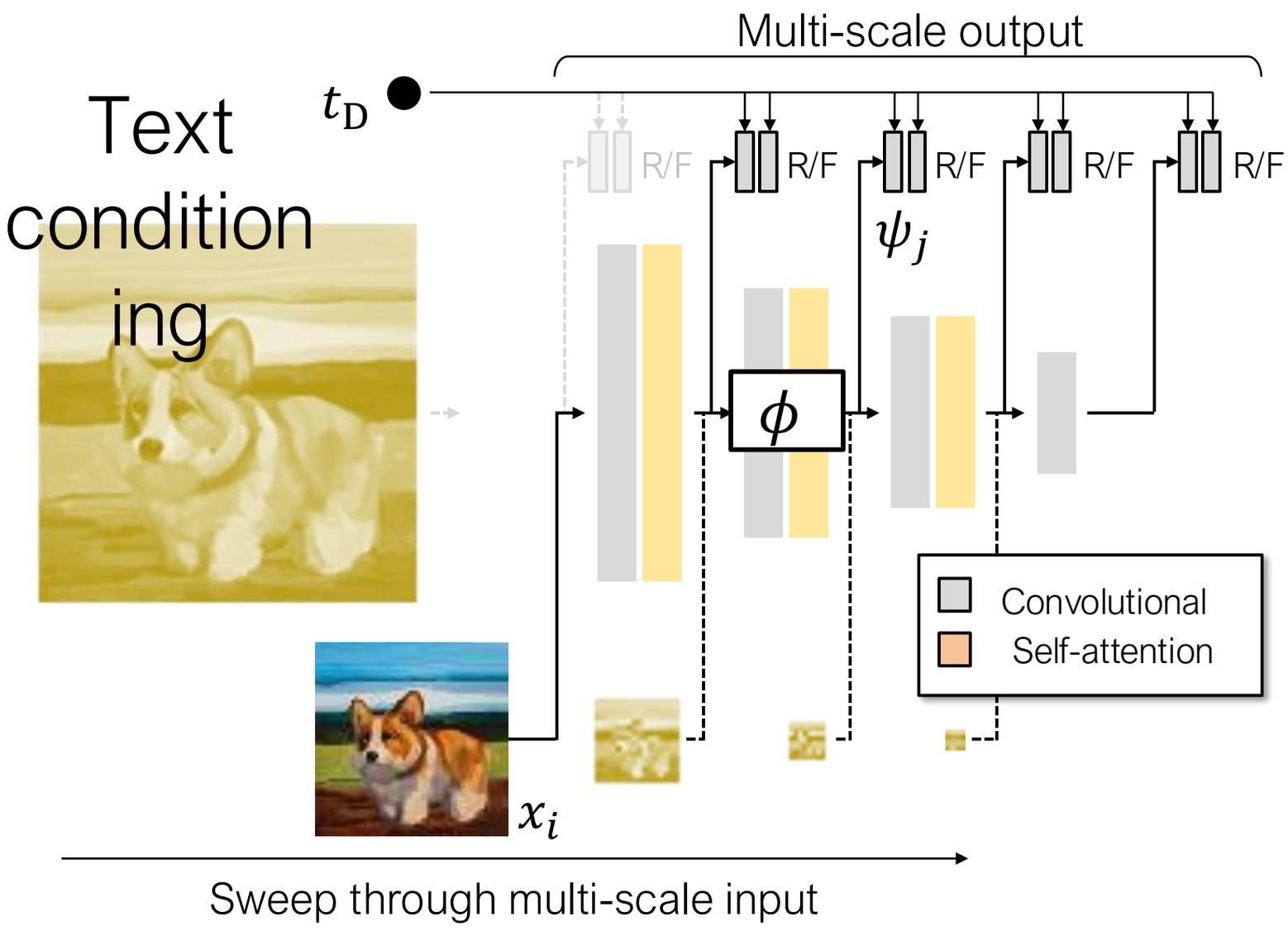
Multi-scale image generator



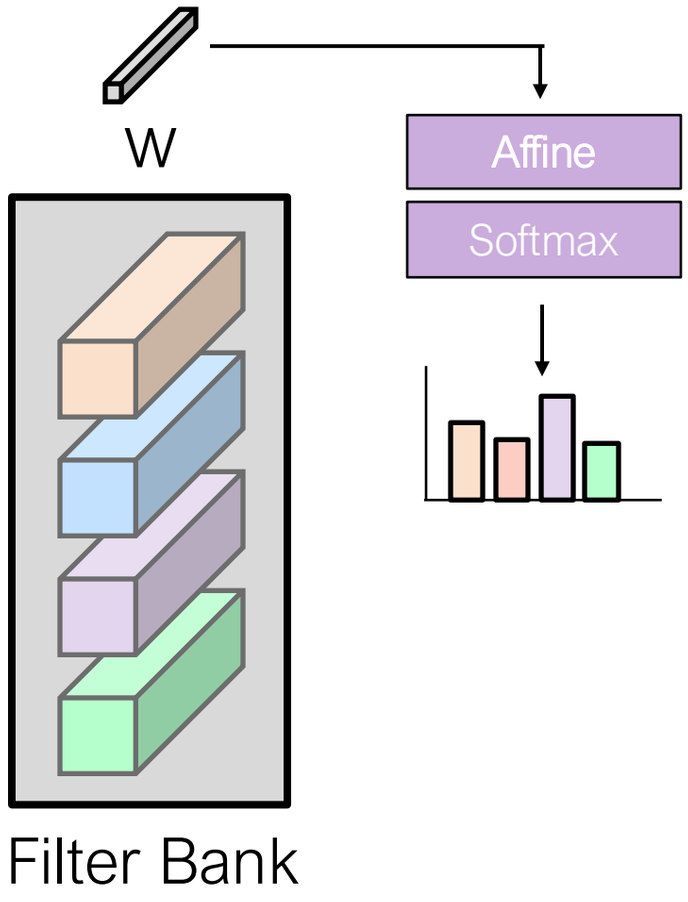
Multi-scale image discriminator



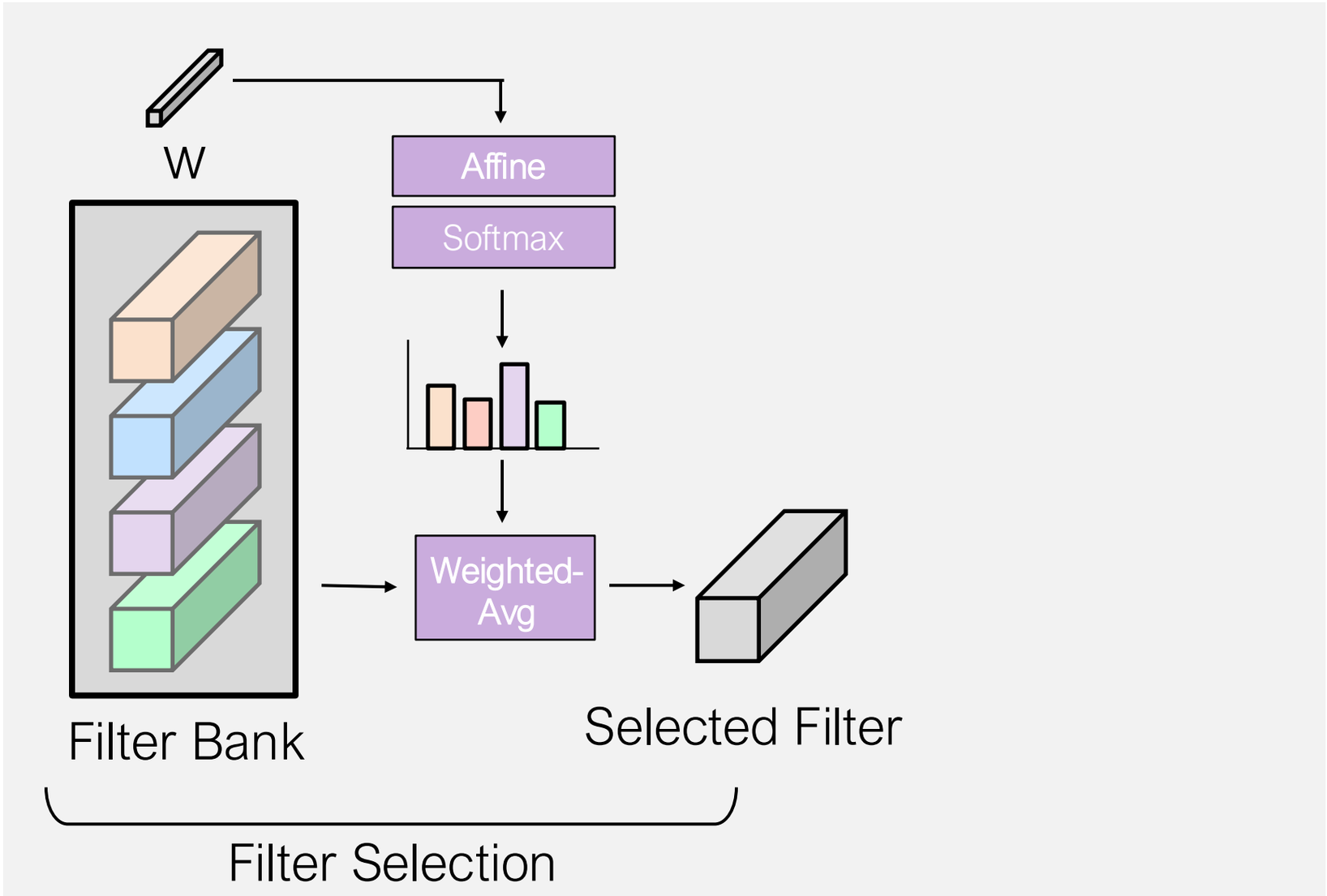
Multi-scale image discriminator



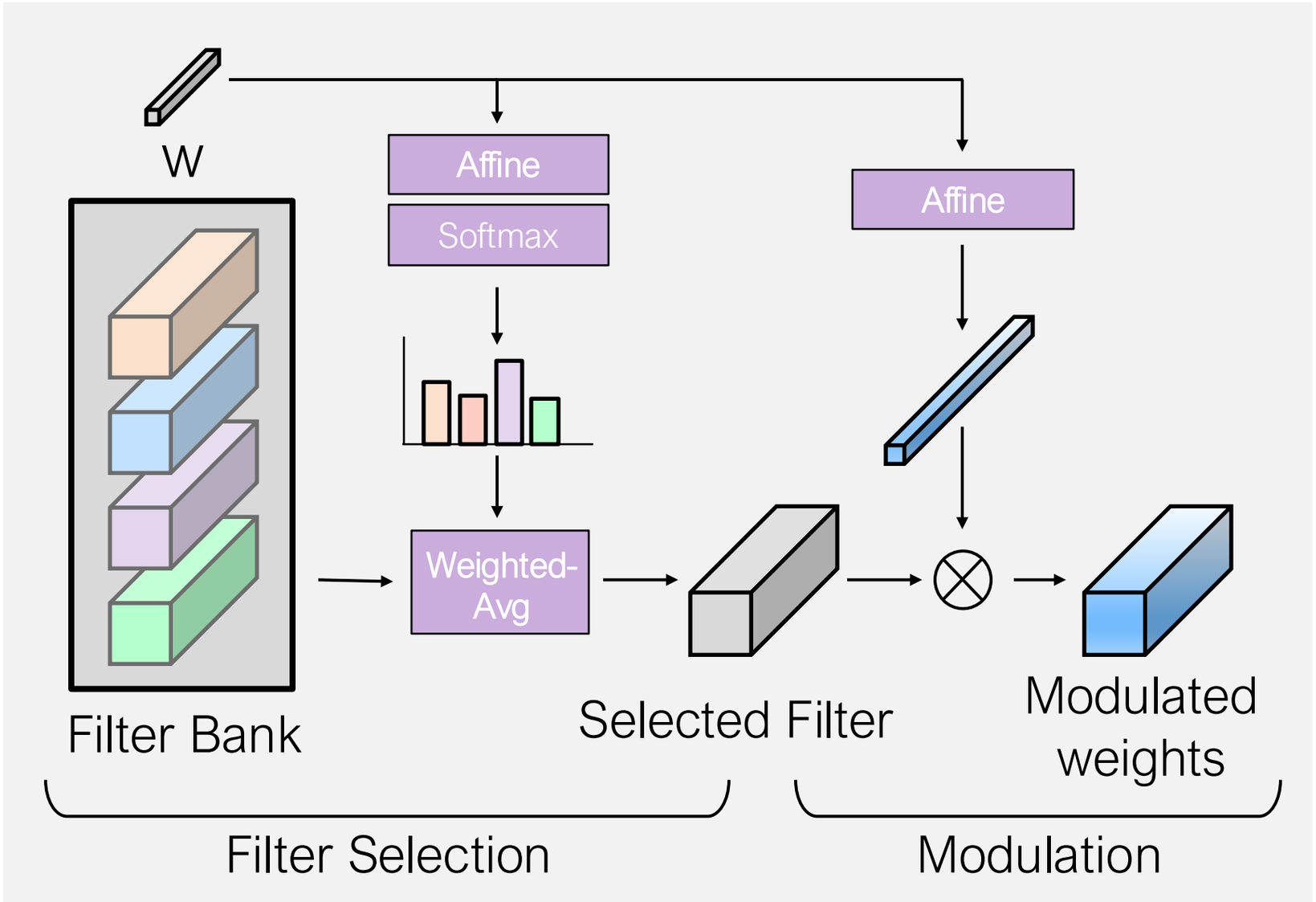
Adaptive Kernel Selection



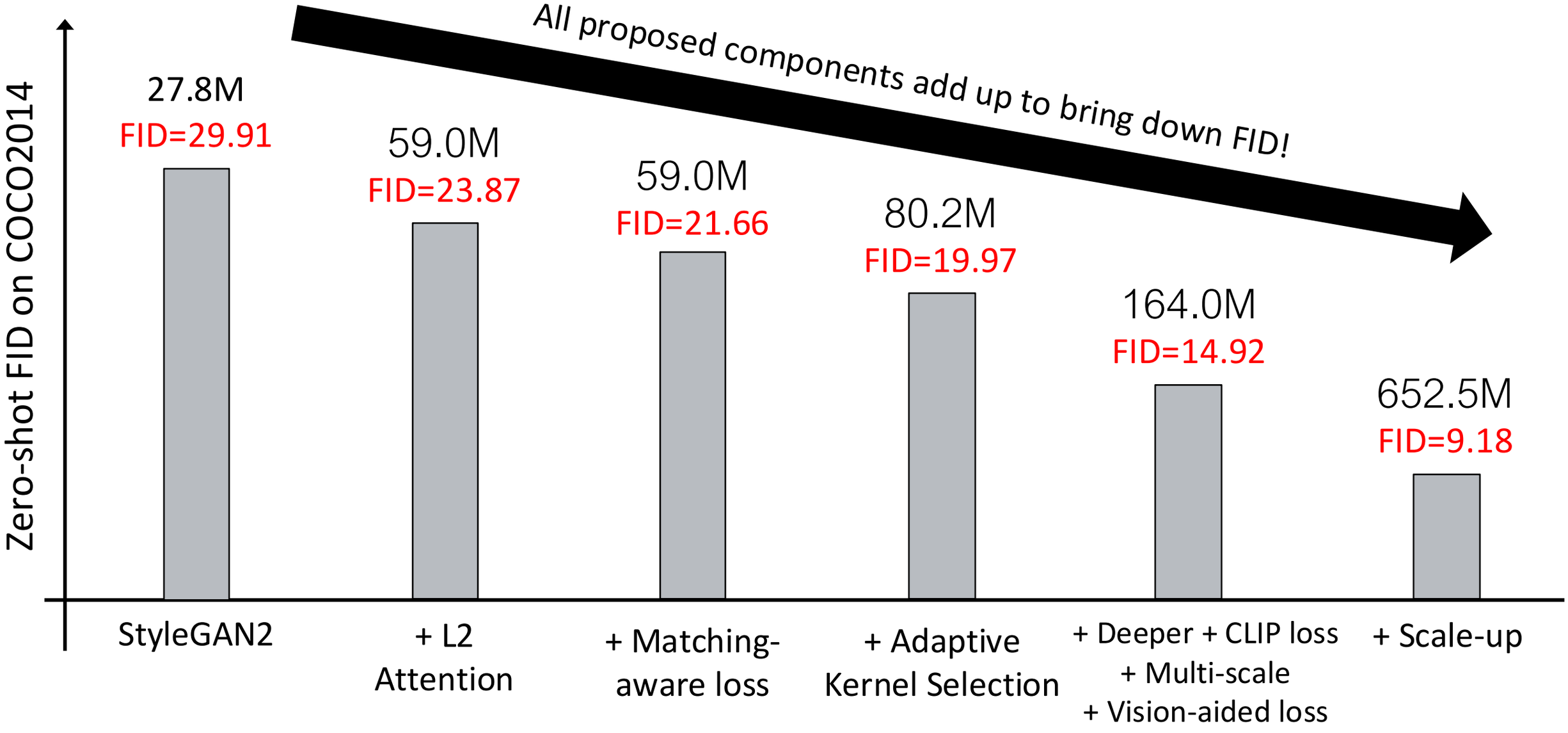
Adaptive Kernel Selection



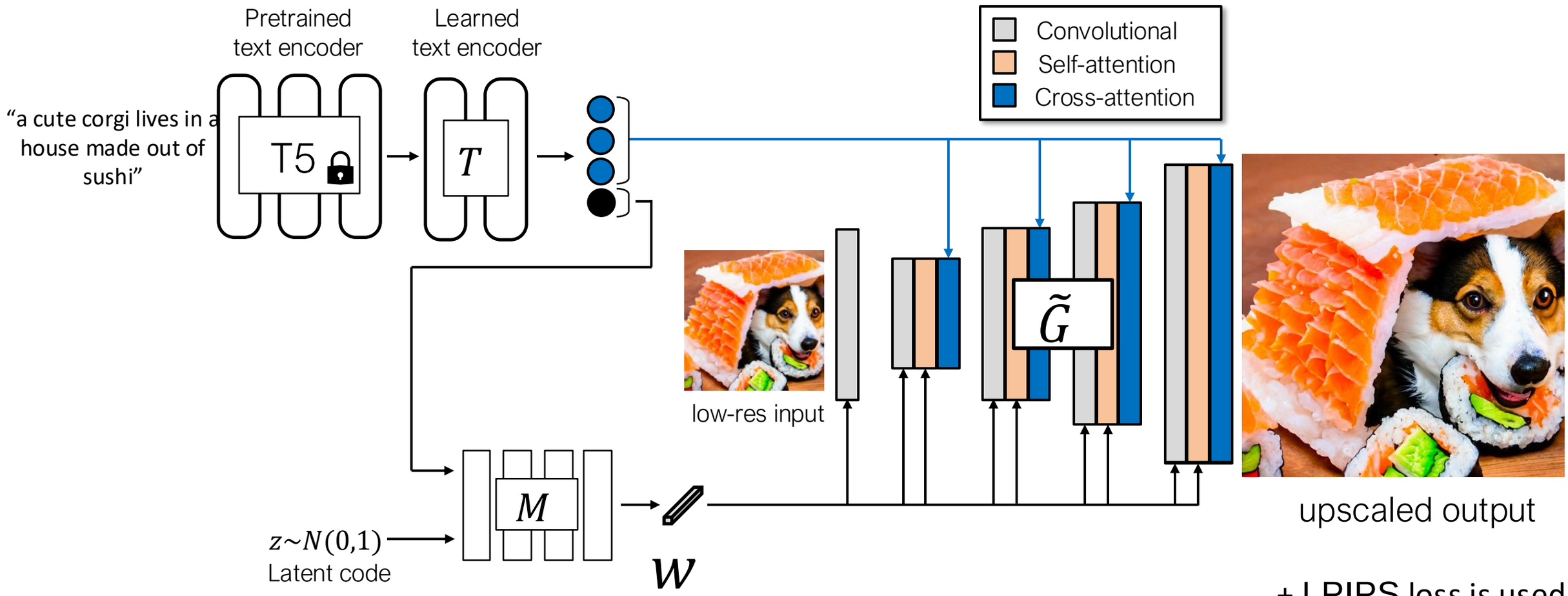
Adaptive Kernel Selection



How we scale up GANs



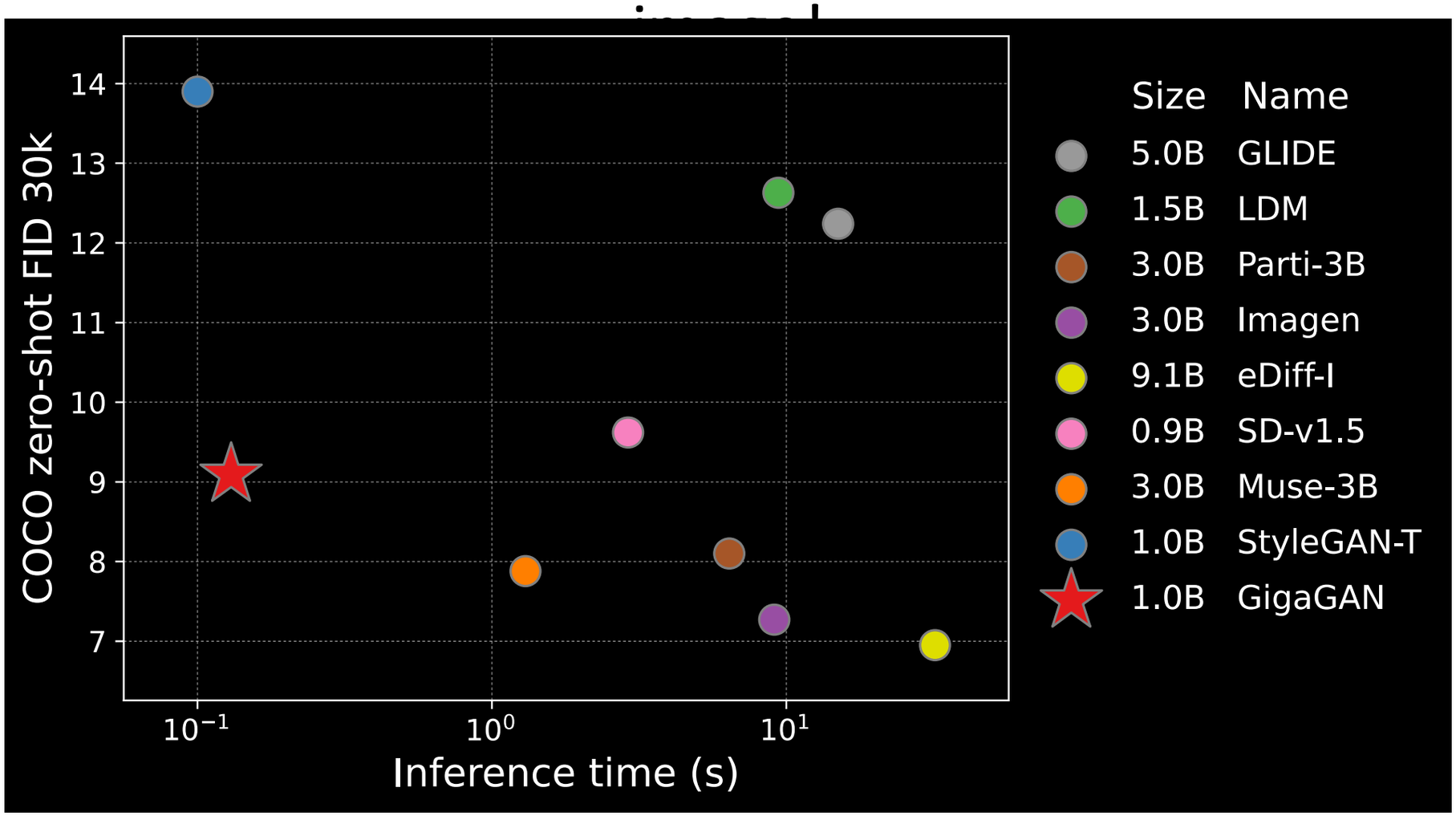
GigaGAN upscaler



+ LPIPS loss is used

GigaGAN achieves FID of 9.09

GigaGAN takes only **0.13s** for generating a 512px

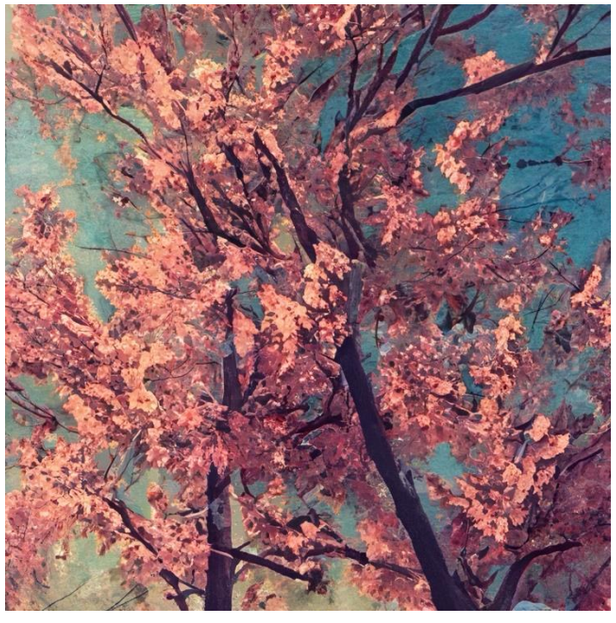


The New York skyline, 4k landscape photography.

Flowers in shape of a heart.

CG art of a majestic castle, evoking a sense of splendor.

Magritte painting of a clock on a beach.

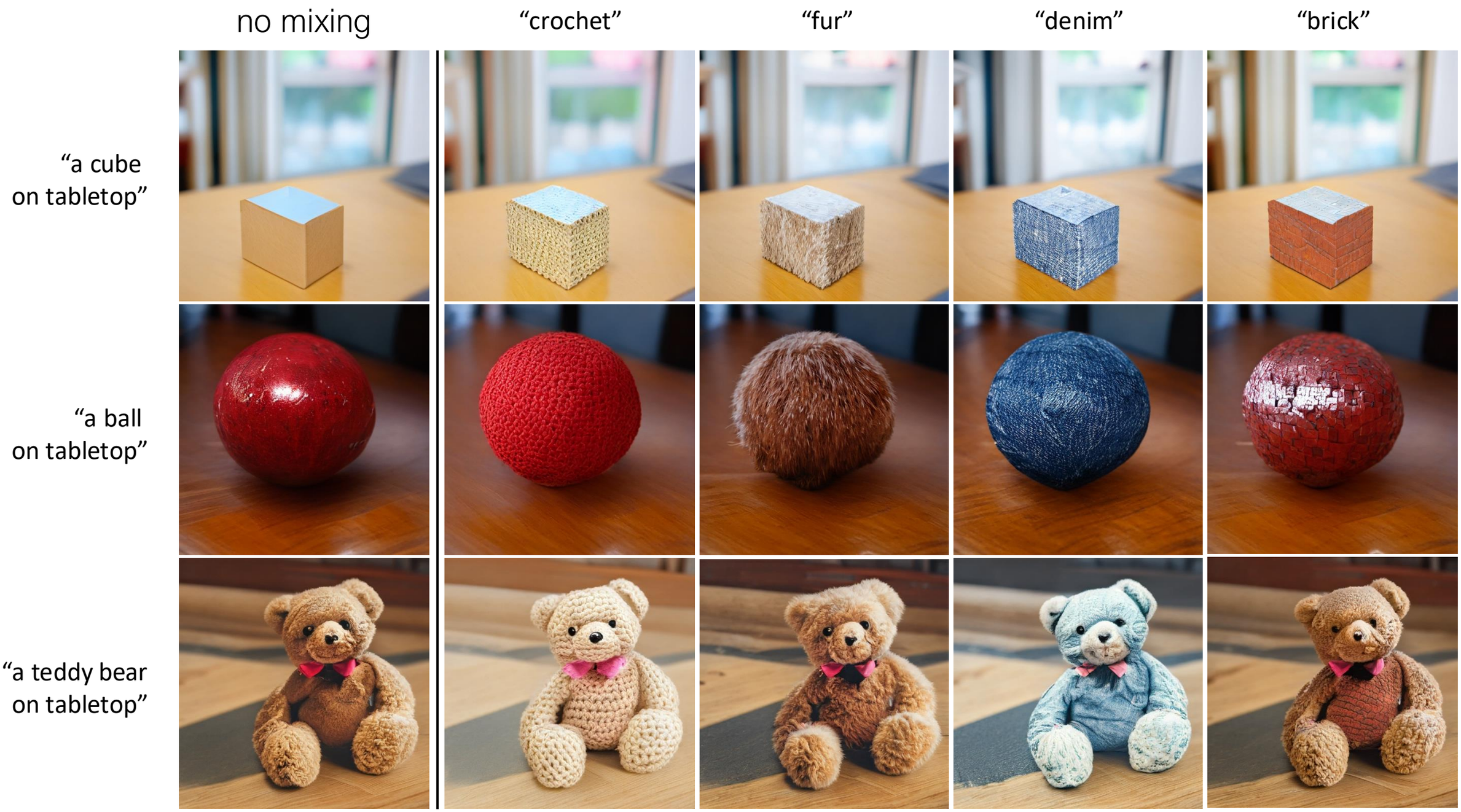


Cherry blossom trees, pastel watercolor style, 2K.

A digital illustration of the Grand Canyon on the moon, depicted in shades of grey.

Smooth dining table with meats in an elegant restaurant with soft lighting.

A futuristic city with a cyberpunk vibe, captured with long camera exposure.


no mixing
“crochet”
“fur”
“denim”
“brick”
“a cube on tabletop”
“a ball on tabletop”
“a teddy bear on tabletop”

Input (128px)

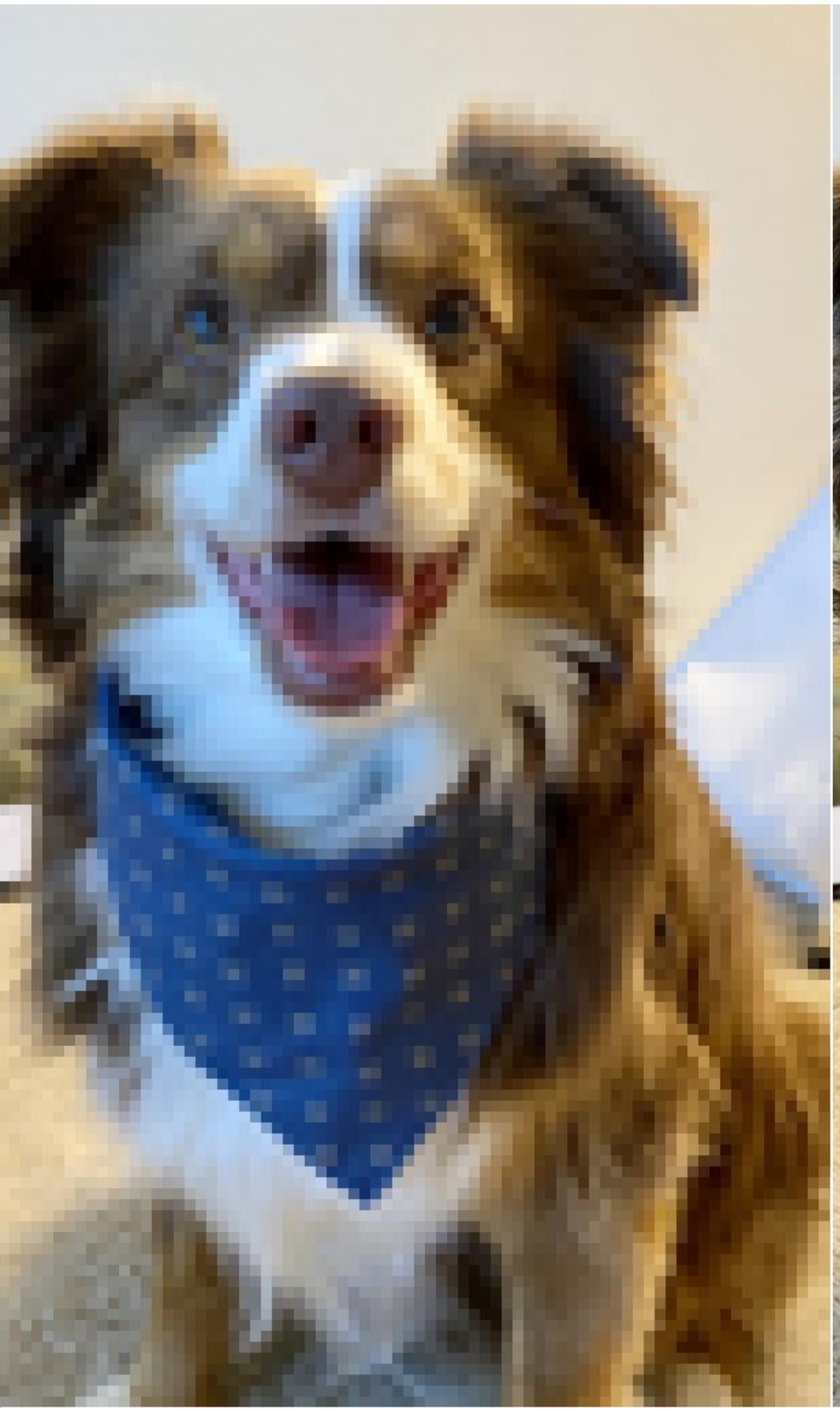
GigaGAN (1024px)

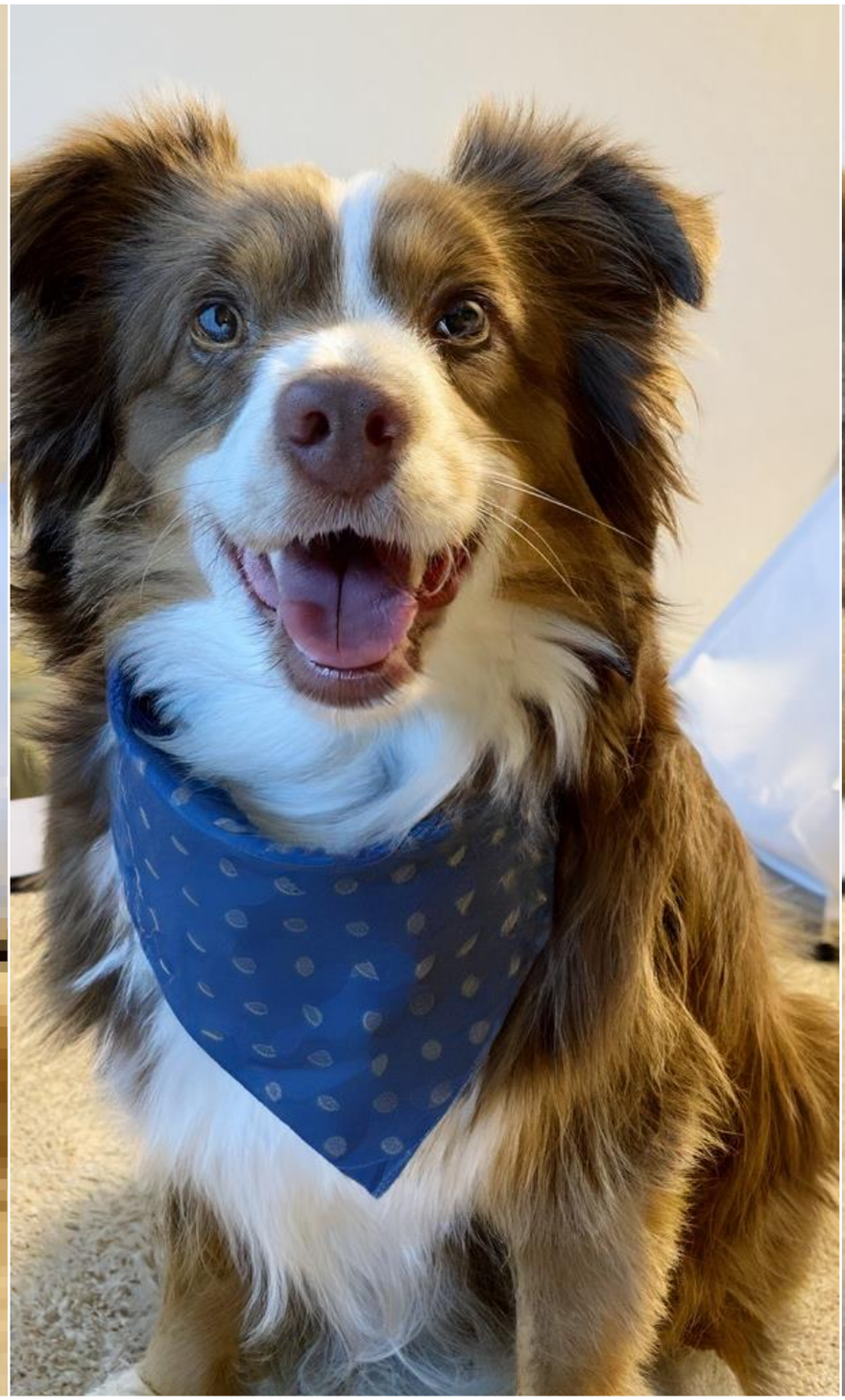
Stable Diffusion (1024px)

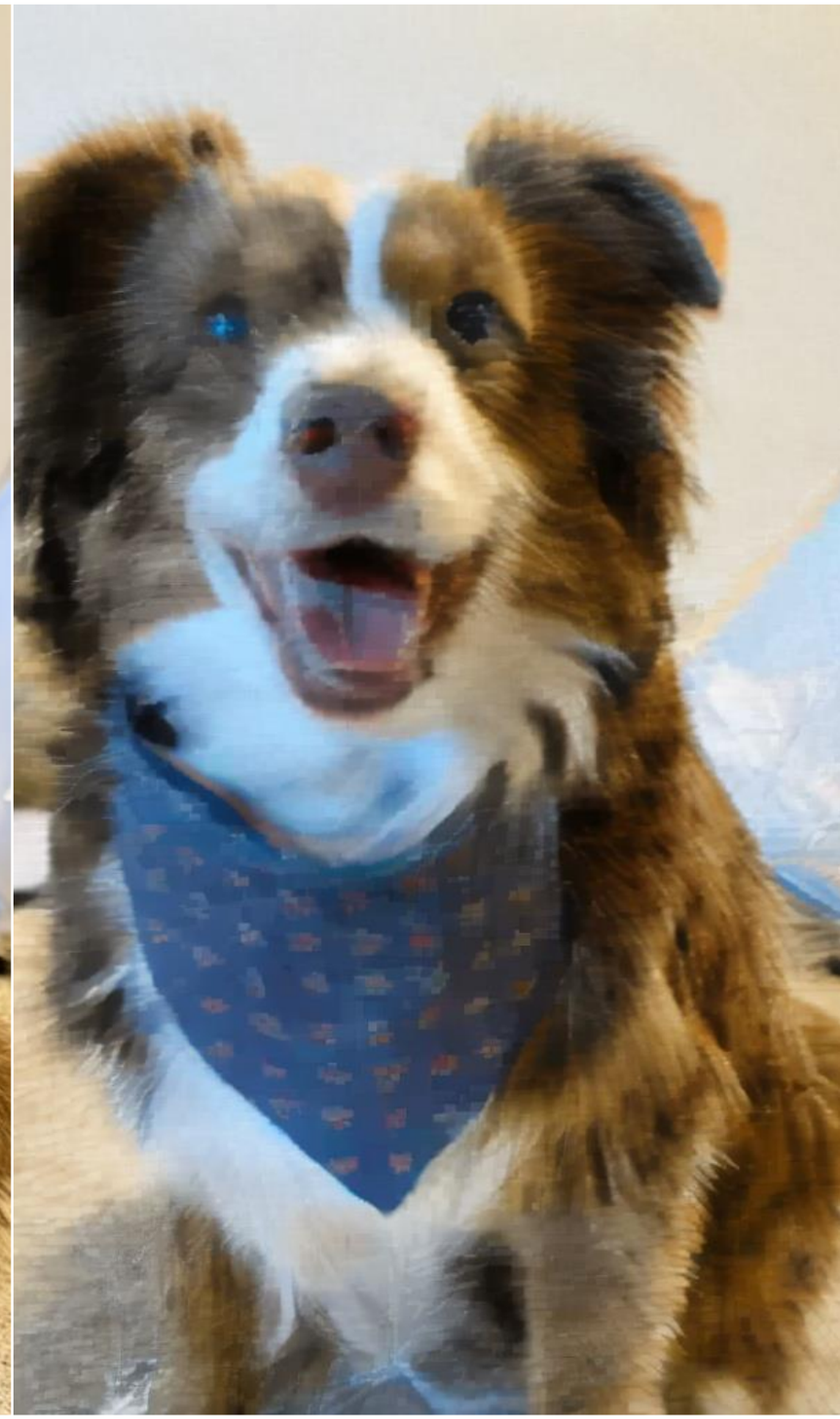
Input (128px)

GigaGAN (1024px)

Stable Diffusion (1024px)







4K Upsampling using GigaGAN

Failure cases

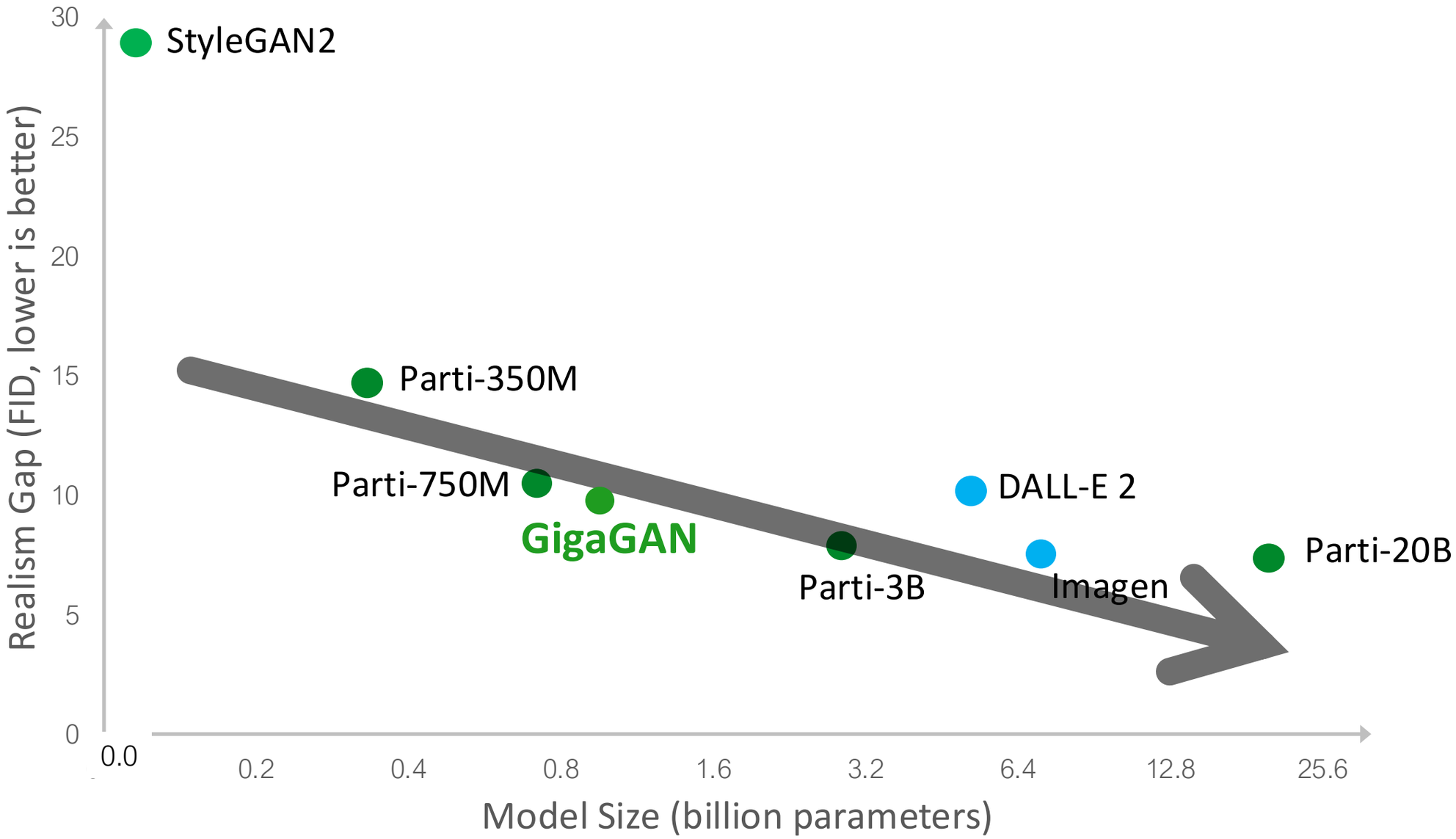
"A teddy bear on a skateboard in times square."

Ours
(0.13s / img)

Stable Diffusion
(2.9s / img)

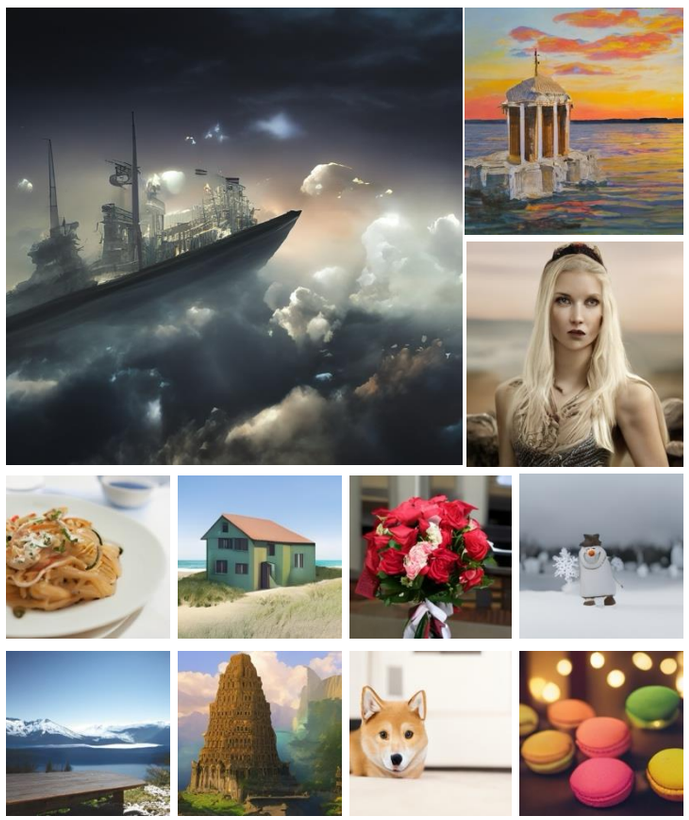
DALL·E 2

Larger models attain better realism



Advantages of GigaGAN

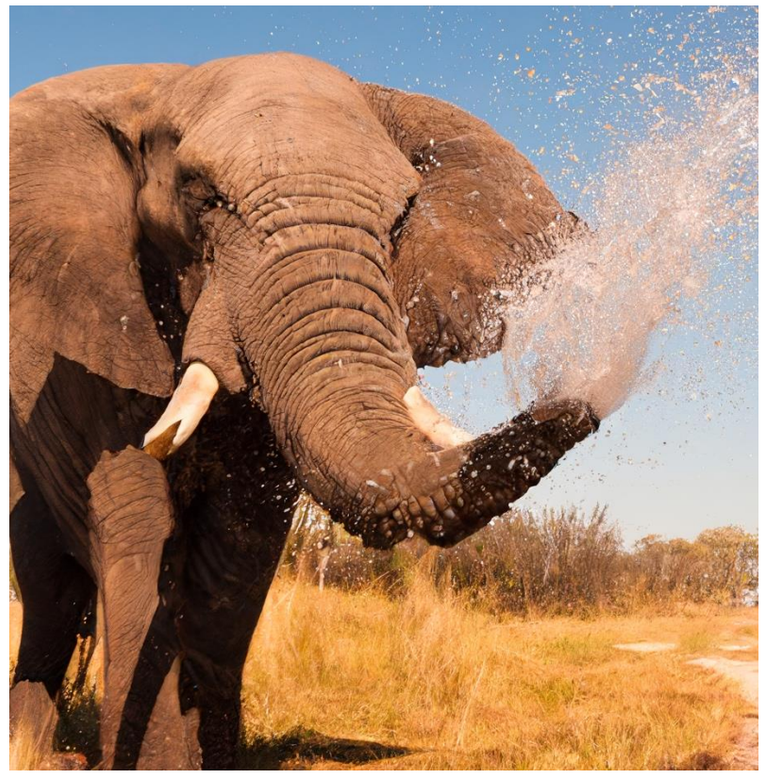
- 0.13s for a 512px image
- 3.7s for a 4096px image



Fast inference

Useful latent space

4K super-resolution

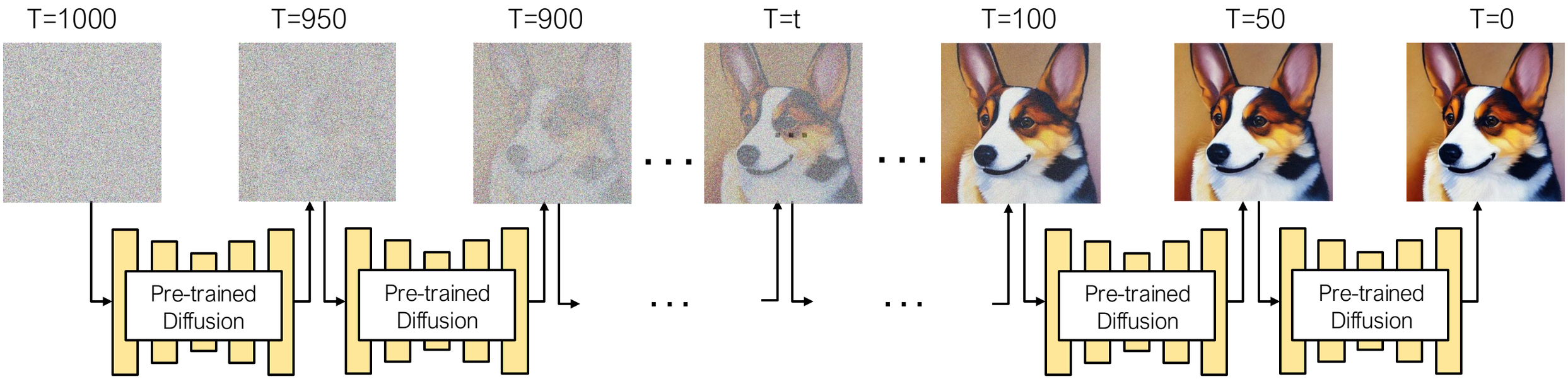


- GigaGAN can be utilized with a modular adversarial loss.

Extensibility to other tasks

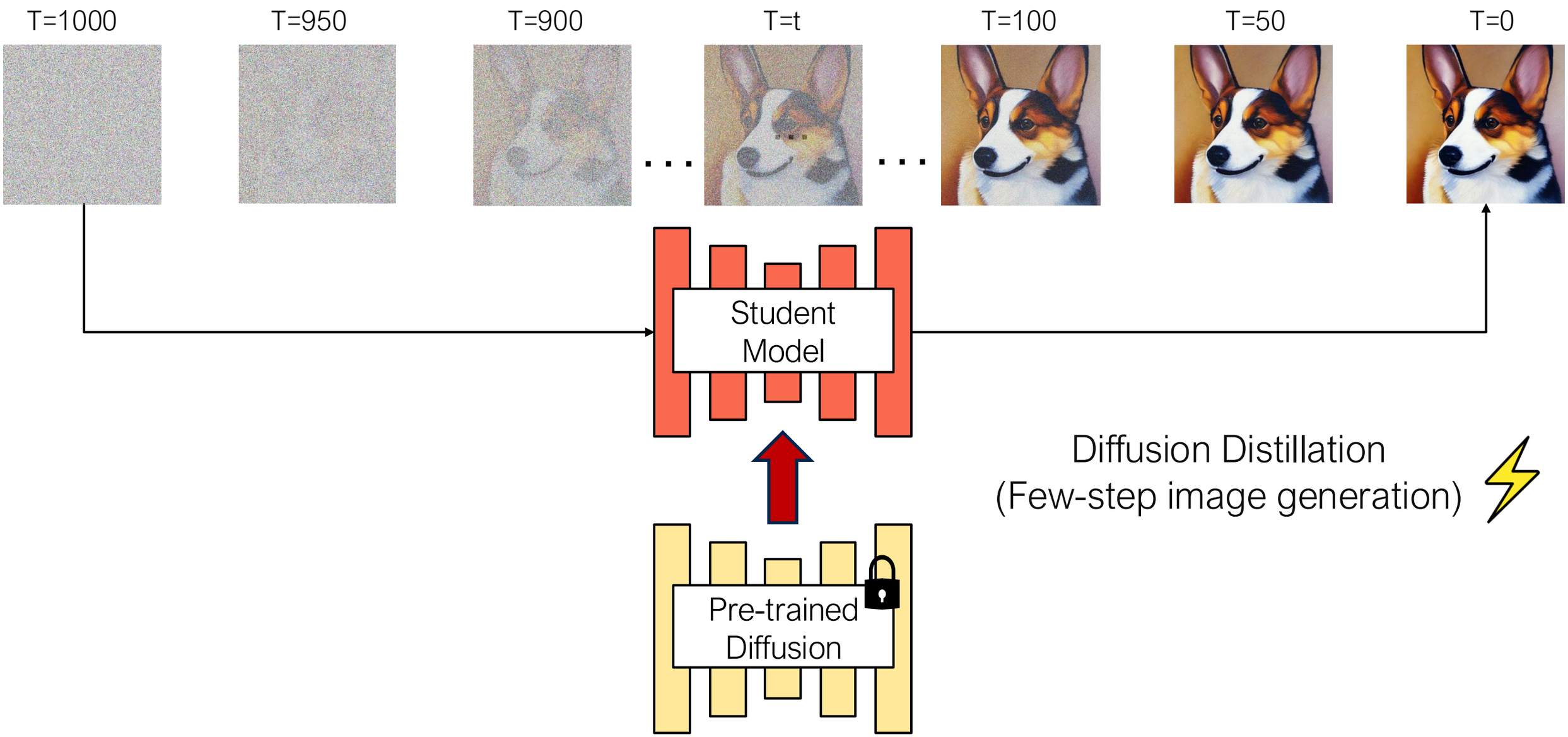
Slow Diffusion Teacher to Fast GAN student

What is diffusion model distillation?



Requires 20~50 steps of denoising to generate a single image.

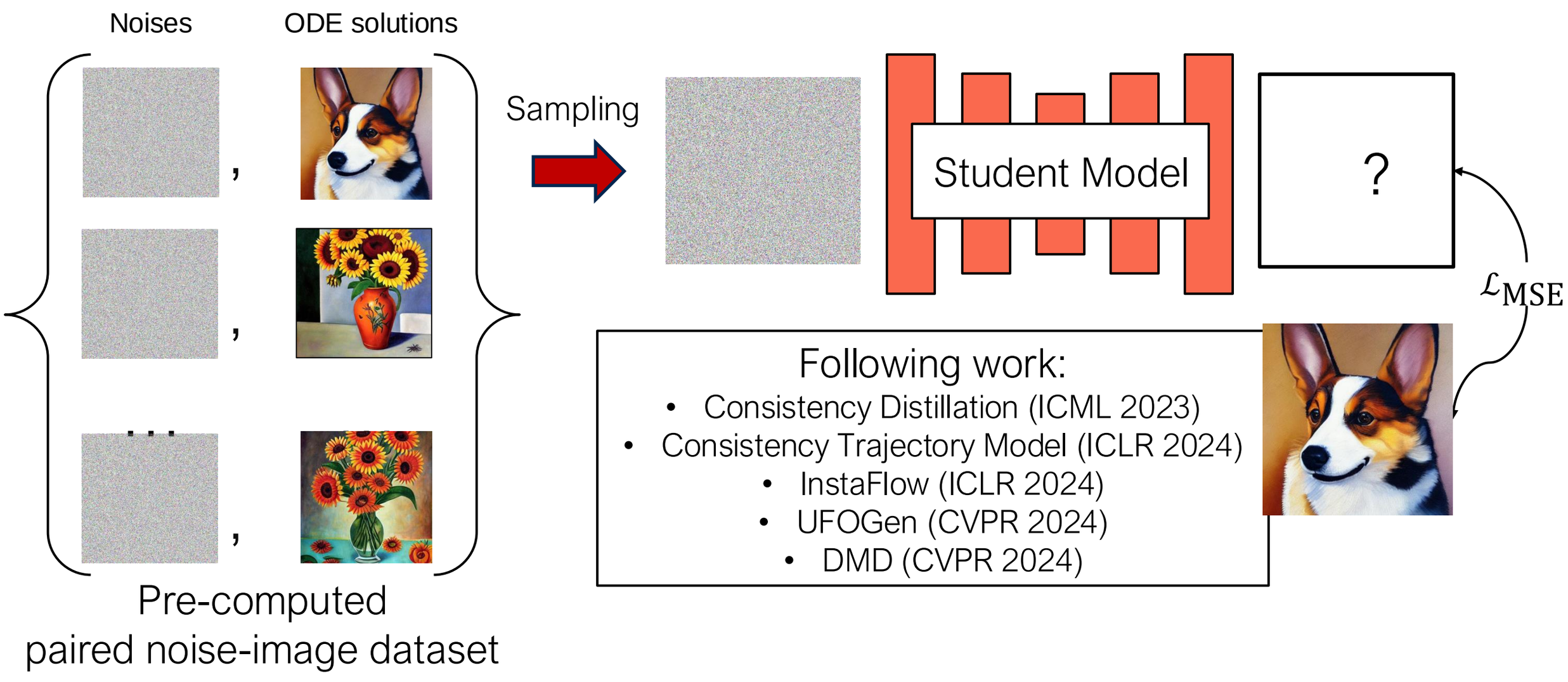
What is diffusion model distillation?



Diffusion Distillation
(Few-step image generation)

How can we distill a diffusion model?

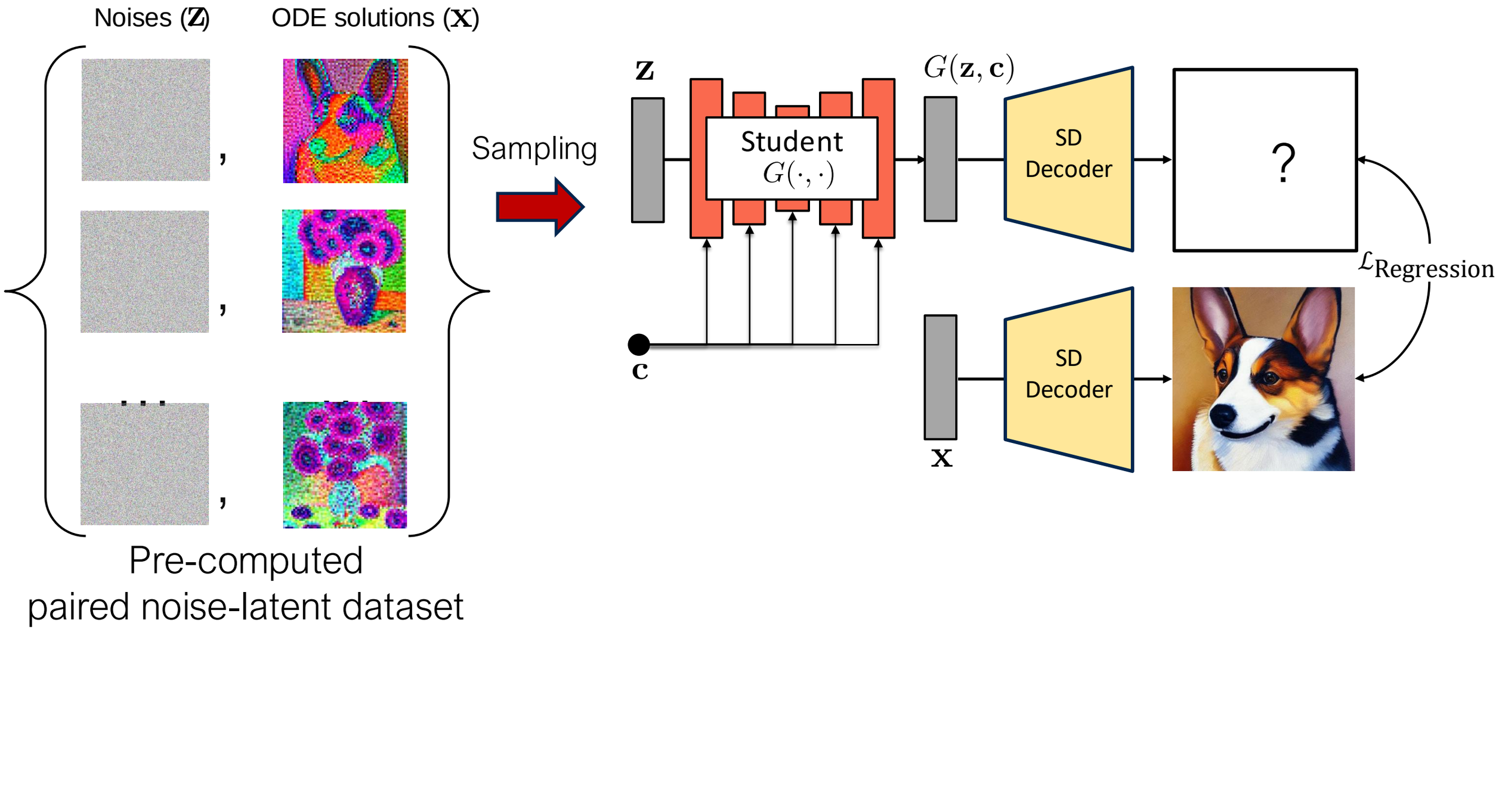
Direct regression approach: Luhman et al., 2021.



Following work:

- Consistency Distillation (ICML 2023)
- Consistency Trajectory Model (ICLR 2024)
- InstaFlow (ICLR 2024)
- UFOGen (CVPR 2024)
- DMD (CVPR 2024)

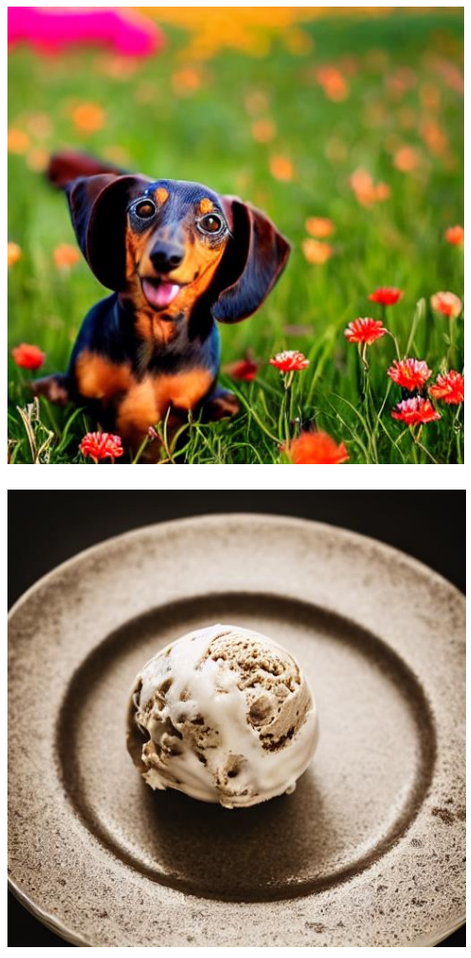
Distillation with Perceptual Loss and GANs



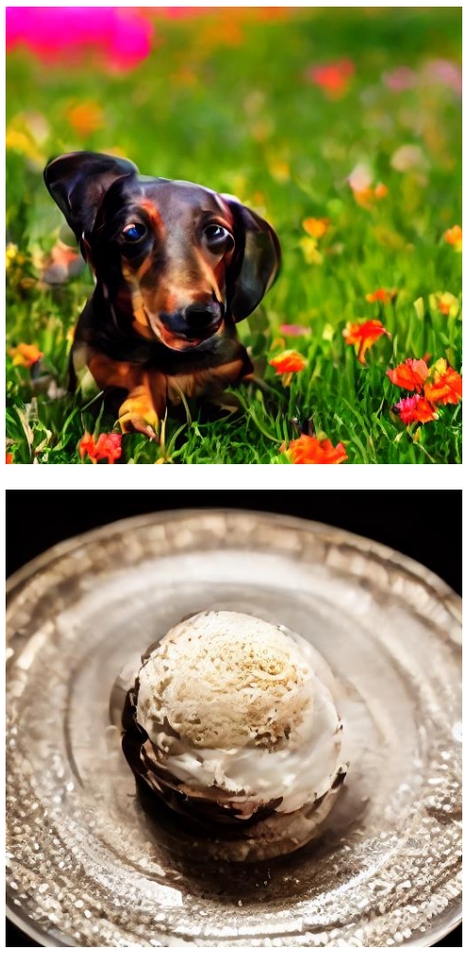
Pre-computed
paired noise-latent dataset

Perceptual loss is what you need

50 DDIM outputs



LPIPS Regression



Loss function	img/sec	FID	CLIP-score
MSE	138.4	110.55	0.222
LPIPS	40.0	25.94	0.288

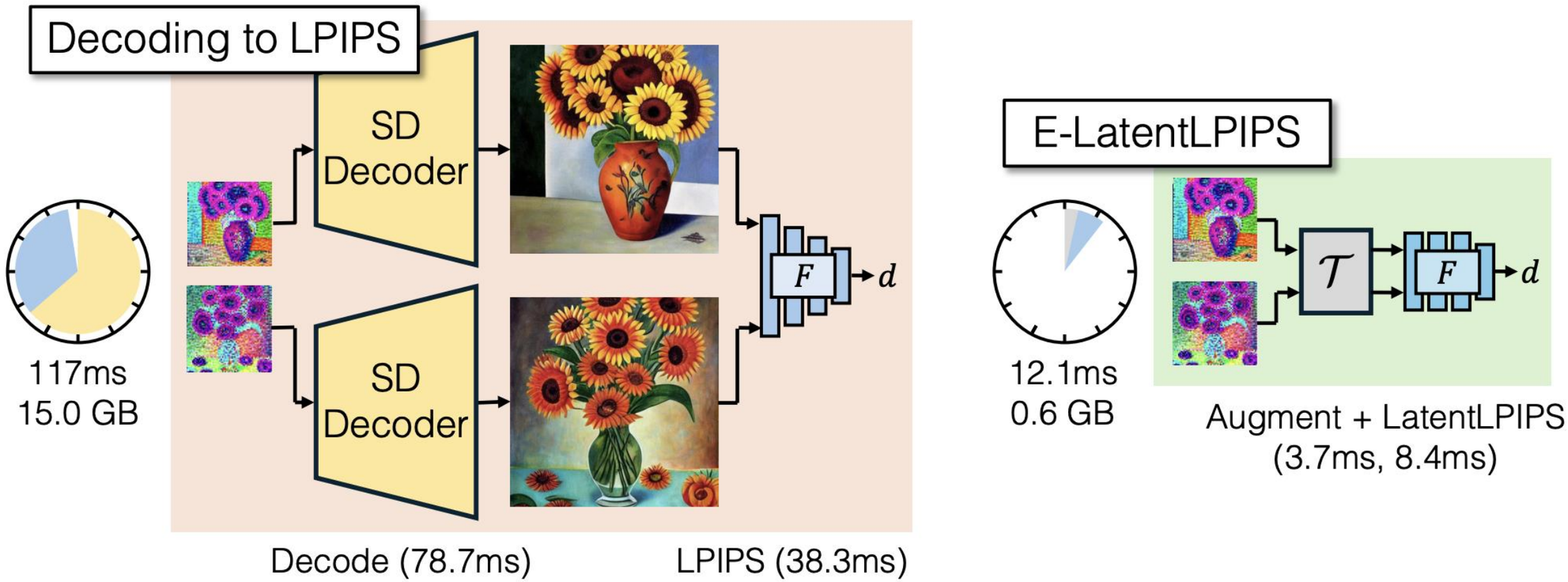
😄 Direct regression using LPIPS = Good

😭 Pixel-space LPIPS is expansive.

😭 Decoding can introduce errors in LPIPS.

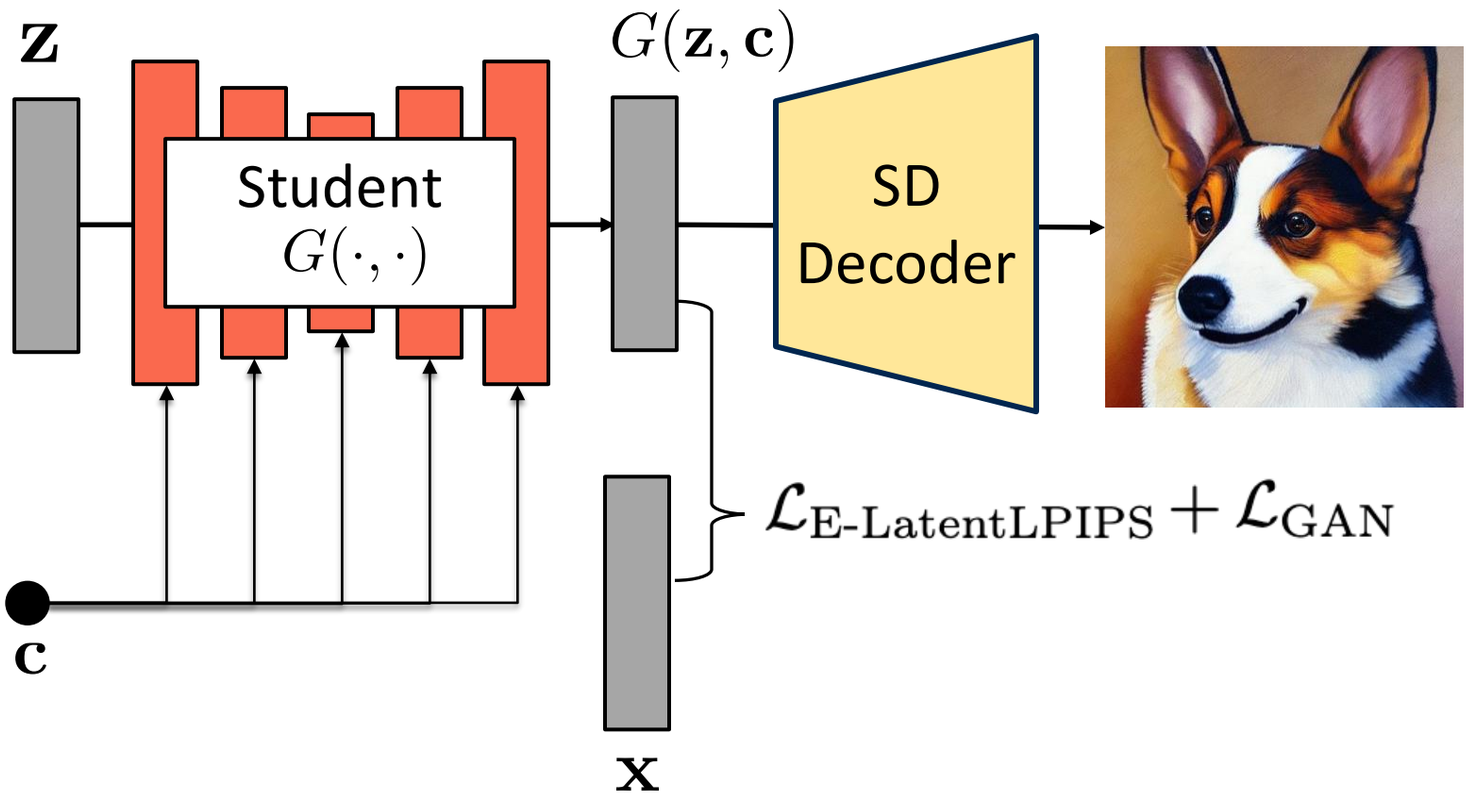
[1] LPIPS: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric, Zhang et al., CVPR, 2018.

Perceptual loss calculation in latent space



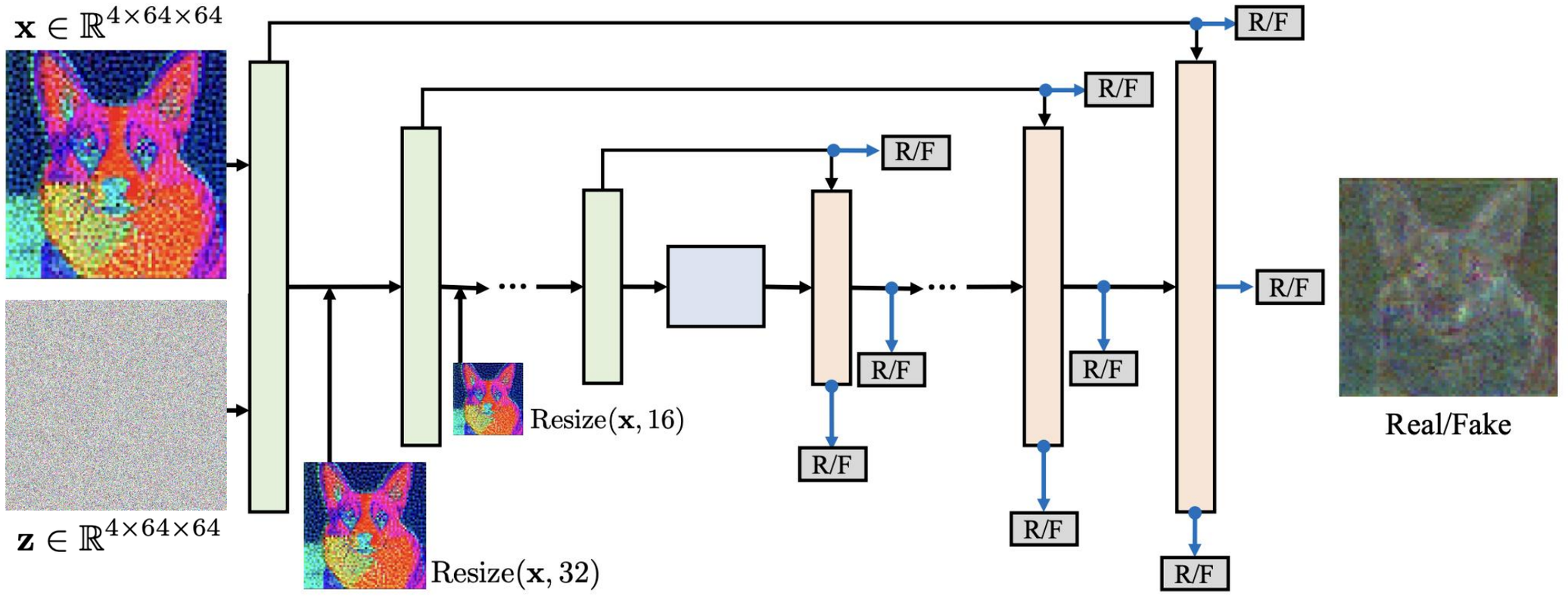
- 9.7× speedup achieved with E-LatentLPIPS.

GAN further improves quality



- In practice, paired image regression yields better results when combined with a conditional GAN loss.
(**conditions** = prompt & noise, **target** = ODE solution)

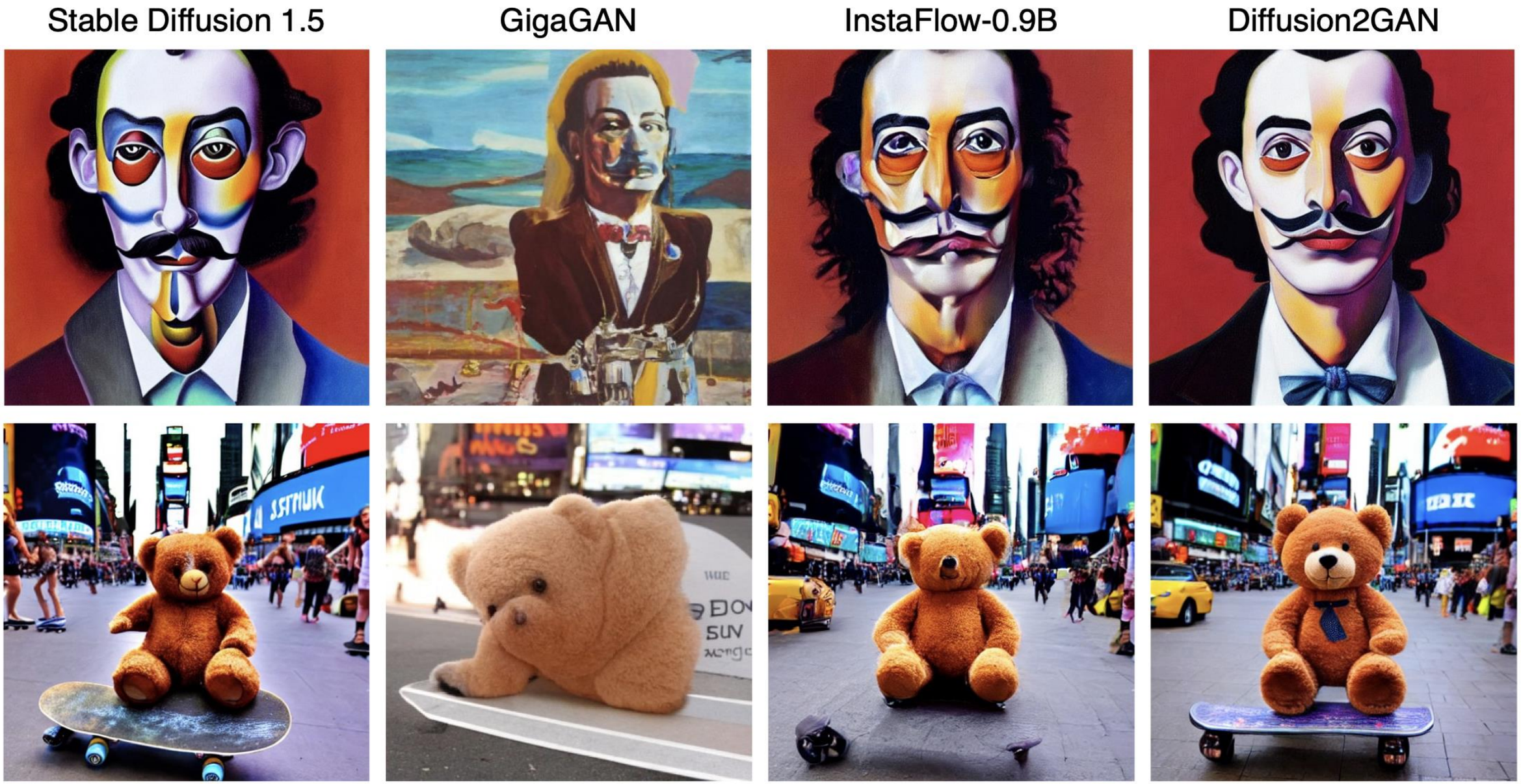
Pre-trained diffusion as a conditional discriminator



- Multi-scale Input/output discriminator following GigaGAN.

[2] Scaling up GANs for Text-to-Image Synthesis, Kang et al., CVPR, 2023.

Visual comparison with previous models



Visual comparison with latest work

SDXL-Turbo (arXiv Nov, 2023)

Diffusion2GAN

"A woman with bright blue eyes and curly blonde hair smiles warmly in a softly lit café."

Visual comparison with latest work

SDXL-Lightening (arXiv Feb., 2024)

Diffusion2GAN

“A woman with bright blue eyes and curly blonde hair smiles warmly in a softly lit café.”

Discussion

Why are GANs hard to train?

Typical answer:

- adversarial training is unstable;
- mode collapse.

Architecture search and scaling:

- Diffusion search space O(N)
- GAN requires G and D: O(N2)

Open-source community

no good open sourced models after StyleGAN series

Multi-step vs. single-step

- Single-step model is harder to train.
- We cannot train single-step AR or Diffusion from scratch.
- GAN is better than L2 and classification loss for single-step.

Why are GANs still useful?

Training image tokenizer (encoder/decoder)

Most of image tokenizers are trained with GAN loss.

Distilling slow models into fast GANs

Conditional Generation (e.g., ControlNet->pix2pix-turbo)

Thank You!